

19.842/14/04



# TUGAS AKHIR

( KP 1701 )

## **PENGUNAAN TEKNIK GEOMETRIC AND TOPOLOGICAL PATTERN MATCHING UNTUK MENENTUKAN LAYOUT RUANG AKOMODASI ABK KAPAL PENUMPANG 1000 ORANG**



RS pe  
623.8243  
Wib  
P-1  
2002

OLEH

**TOTOK HARI WIBOWO**  
NRP. 4197 100 002

**JURUSAN TEKNIK PERKAPALAN  
FAKULTAS TEKNOLOGI KELAUTAN  
INSTITUT TEKNOLOGI SEPULUH NOPEMBER  
SURABAYA**

**2002**

PERPUSTAKAAN ITS	
Tgl. Terima	8-4-2003
Terima Dari	H1
No. Agenda Prp.	216839

# LEMBAR PENGESAHAN

## PENGUNAAN TEKNIK

### *GEOMETRIC AND TOPOLOGICAL PATTERN MATCHING* UNTUK MENENTUKAN LAYOUT RUANG AKOMODASI ABK KAPAL PENUMPANG 1000 ORANG

## TUGAS AKHIR

Telah Direvisi Sesuai dengan Hasil Sidang Ujian Tugas Akhir

pada

Jurusan Teknik Perkapalan

Fakultas Teknologi Kelautan


Institut Teknologi Sepuluh Nopember

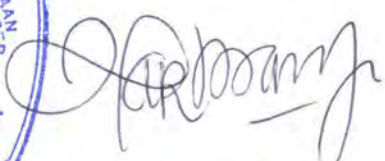
Surabaya

Mengetahui / Menyetujui :

Dosen Pembimbing I

Dosen Pembimbing II

  
I G. N. S. Buana, S.T., M.Eng.  
NIP. 132 085 800

  
Ir. Djauhar Manfaat, M.Sc., Ph.D.  
NIP. 131 651 444

**SURABAYA**  
**2002**

## ABSTRAK

Bagian dari suatu proses perencanaan kapal adalah pembuatan Rencana Umum. Dan inti dari pembuatan Rencana Umum adalah penentuan ruang-ruang akomodasi dan penempatannya pada geladak akomodasi. Permasalahan yang akan timbul adalah jika seorang desainer merencanakan tata letak ruangan tersebut dari nol (*from scratch*).

Cara cepat merencanakan tata letak ruang akomodasi tersebut yang sesuai dengan keinginan pemesan (*owner*) adalah dengan memanfaatkan desain lama (*design reuse*). Pendekatan yang memanfaatkan *design reuse* untuk pembuatan desain baru adalah pendekatan *Case Based Reasoning* (CBR) (Kolodner 1993). Untuk memecahkan permasalahan desain tata letak ruang akomodasi dipakai pendekatan *Case Based Reasoning* (CBR) *in Design* (Maher dkk. 1995). Dalam penarikan (*recall*) desain lama ini digunakan teknik *geometric and topological pattern matching* yaitu mencari kesamaan desain tata letak ruang akomodasi yang baru dengan desain lama yang ada di *case base* berdasarkan tingkat kesamaan bentuk geometri dan kedekatan antar ruang.

Pemanfaatan komputer akan lebih mempermudah dalam penarikan dan akan memberikan urutan desain dari yang mempunyai tingkat kesamaan bentuk dan kedekatan tertinggi sampai terendah, selanjutnya desain dengan kesamaan tertinggi yang akan dijadikan sebagai desain baru sesuai dengan keinginan pemesan. Dengan mempertimbangkan keuntungan tersebut, maka semua teknik diatas diimplementasikan kedalam sebuah program komputer yang disebut *Sistem SpaceMan*.



## ABSTRACT

The important part of a ship design process is planning General Arrangement. And the principle of planning General Arrangement is how to specify the position of a layout accommodation and its replacement on a deck accommodation. The problem will appear if a designer creates layout accommodation design from scratch.

The fast way to solve the design of layout accommodation that fulfill the owner's design requirement is using past design cases (design reuse) for new design process. An approach commonly uses design reuse to create new design is *Case Based Reasoning* (CBR) approach (Kolodner 1993). To solve the design layout accommodation problem is used *Case Based Reasoning* (CBR) in Design (Maher et al., 1995) approach. In recalling past design cases, the writer use *geometric and topological pattern matching technique* to find the layout similarity between the owners requirement and past design in case base, which is viewed from similarities adjacencies of one space to another and geometric shape.

The using computer can make easier in recalling and give the similarity value of geometric and topological from high level to low level of similarity, and the past design has high level of similarity will be template of new design. Consider to the advantages, all technique above are implemented to a computer program called *SpaceMan System*.



## KATA PENGANTAR

Syukur Alhamdulillah penulis panjatkan kepada Allah SWT yang telah memberi rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan Tugas Akhir ini tepat pada waktunya.

Semoga Tugas Akhir ini memberi manfaat pada dunia desain, khususnya Bidang Perkapalan. Karena Tugas Akhir semacam ini merupakan wacana baru yang dapat memperluas pengetahuan kita khususnya dalam Bidang Perkapalan. Penulis berharap Tugas Akhir ini menjadi jendela yang menunjukkan bahwa di dunia ini banyak ilmu dan pengetahuan yang belum kita ketahui yang dapat diterapkan dalam Bidang Perkapalan.

Sebagai manusia biasa penulis tidak akan mampu menyelesaikan Tugas Akhir ini tanpa bantuan dan dukungan dari berbagai pihak. Untuk itu penulis menyampaikan terima kasih yang sebesar-besarnya kepada:

1. Ir. Djauhar Manfaat, M.Sc., Ph.D., selaku dosen pembimbing dan orang yang sangat menentukan sekali terhadap selesainya Tugas Akhir ini.
2. I G. N. Sumanta Buana, S.T., M.Eng., selaku pembimbing yang mempunyai kepedulian yang besar terhadap penulis dalam penyelesaian Tugas Akhir ini.
3. M. Thohir, Wahyu Wiyati dan Wiralina Handayani, orang-orang yang berperan dalam mengawali (*starting*) Tugas Akhir ini.

4. Habibur Rohman, Dian Lisza dan Arif Yunaedi, orang-orang sebasib sepenanggungan, selalu bersama dalam suka dan duka dalam penyelesaian (*finishing*) Tugas Akhir ini.
5. Teman-teman angkatan '97, serta *crew* Lab. Sistrans baik secara langsung maupun tidak langsung memberi dukungan terhadap penyelesaian Tugas Akhir ini.
6. Hasyim dan Lutfi, serta para Pimpinan Lab. Sistrans yang telah rela menyediakan sarana dan prasarana dalam penyelesaian Tugas Akhir ini.

Penulis sadar bahwa penulisan Tugas Akhir ini masih ada kekurangannya, maka saran dan kritik yang bersifat membangun sangat diharapkan. Semoga Tugas Akhir ini dapat dimanfaatkan untuk hal yang positif sesuai dengan tujuan dari penulisan Tugas Akhir ini.

Surabaya, Agustus 2002

**Penulis**

## DAFTAR ISI

	Halaman
JUDUL TUGAS AKHIR	
LEMBAR PENGESAHAN	
ABSTRAK .....	iv
KATA PENGANTAR .....	vi
DAFTAR ISI .....	viii
DAFTAR GAMBAR .....	xi
DAFTAR TABEL .....	xiii
BAB I     PENDAHULUAN	
1.1. Latar Belakang Masalah .....	1 - 1
1.2. Perumusan Masalah .....	1 - 3
1.3. Tujuan dan Manfaat .....	1 - 3
1.4. Batasan Masalah .....	1 - 4
1.5. Metodologi dan Model Analisis .....	1 - 5
1.6. Sistematika Laporan .....	1 - 7
BAB II    TINJAUAN PUSTAKA	
2.1. Rencana Umum .....	2 - 1
2.1.1. Ruang .....	2 - 2
2.1.2. Ruang Akomodasi .....	2 - 4
2.1.3. Ruang Akomodasi ABK .....	2 - 7
2.1.4. Tata Letak Ruang Akomodasi Kapal .....	2 - 8



2.2. Pemanfaatan Desain Lama ( <i>Design Reuse</i> ) .....	2 - 10
2.3. Pemanfaatan Komputer Untuk Menunjang	
<i>Design Reuse</i> .....	2 - 11
2.4. <i>Case Based Reasoning</i> .....	2 - 13
2.5. <i>Case Based Reasoning in Design</i> .....	2 - 15
2.5.1. Representasi Design Cases .....	2 - 19
2.5.1.1. Isi dari <i>design case</i> .....	2 - 19
2.5.1.2. Representasi dari isi <i>design case</i> .....	2 - 21
2.5.1.3. Pengorganisasian <i>design case</i> dalam	
memori <i>case base</i> .....	2 - 23
2.5.2. Penarikan <i>Design Case</i> .....	2 - 24
2.5.2.1. <i>User-directed case selection</i> .....	2 - 24
2.5.2.2. <i>Pattren matching-based layout case</i>	
<i>retrieval</i> .....	2 - 25
2.5.3. <i>Case Adaptation</i> .....	2 - 25
2.6. Pendekatan <i>Pattern Matching</i> .....	2 - 27
2.6.1. Teknik-teknik pattern matching .....	2 - 27
2.6.2. Metode matching .....	2 - 30
2.7. Sistem SPIDA .....	2 - 33
 BAB III   TEKNIK IMPLEMENTASI	
3.1. Representasi Ruang .....	3 - 1
3.2. Representasi Layout .....	3 - 3
3.3. <i>Topological Pattern Matching</i> .....	3 - 4

3.4. <i>Geometric Pattern Matching</i>	3 - 7
3.5. Representasi Bentuk	3 - 8
3.6. Penyusunan Data	3 - 9
3.7. Prosedur Pengolahan Data	3 - 10
3.8. Implementasi pendekatan <i>Case Based Reasoning</i> <i>in Design</i>	3 - 13
BAB IV SISTEM SPACEMAN	
4.1. Sistem SpaceMan	4 - 1
4.2. <i>Library</i>	4 - 3
4.3. Penyamaan Pola ( <i>Panttern Matcher</i> )	4 - 3
4.4. <i>Layout Viewer</i>	4 - 3
4.5. Tampilan Window ( <i>Interface</i> )	4 - 4
BAB V ANALISA HASIL	5 - 1
BAB VI KESIMPULAN DAN SARAN	
6.1. Kesimpulan	6 - 1
6.2. Saran	6 - 1
DAFTAR PUSTAKA	
LAMPIRAN	



## DAFTAR GAMBAR

			Halaman
Gambar	1.1	Diagram metodologi dan analisis .....	1 - 6
Gambar	2.1	Model sederhana dari <i>Case Based Reasoning</i> (CBR) (Maher dkk. 1995) .....	2 - 18
Gambar	3.1	Grepresentasi bangun persegi panjang (a), dan poligon (b) .....	3 - 1
Gambar	3.2	Representasi sebuah graph (Ballard dan Brown, 1982) .....	3 - 5
Gambar	3.3	Dua layout sederhana untuk pembuatan suatu <i>assosiation graph</i> .....	3 - 6
Gambar	3.4	Suatu <i>assosiation graph</i> yang dihasilkan dari <i>matching</i> dua layout pada gambar 3.3 .....	3 - 6
Gambar	3.5	(a) Dua layout sederhana yang identik untuk <i>geometric shape matching</i> (b) <i>Association graph</i> dari dua layout .....	3 - 7
Gambar	3.6	Partisi bentuk: (a) Seatu bentuk terpartisi dalam segitiga; (b) Hasil <i>binary tree</i> dari partisi pada gambar (a) (Leu dan Huang, 1988) .....	3 - 8
Gambar	3.7.	Gambar dua ruang yang didefinisikan kedekatannya .....	3 - 12
Gambar	3.7.	Diagram <i>recalling process</i> .....	3 - 14



Gambar	4.1	Arsitektur dari Sistem SpaceMan .....	4 - 2
Gambar	4.2	Interface utama sistem SpaceMan .....	4 - 4
Gambar	4.3	Menu pilihan dari tombol <b>Deck Cases</b> .....	4 - 5
Gambar	4.4	Dialog untuk menampilkan pilihan layout .....	4 - 5
Gambar	4.5	Pilihan layout untuk semua data layout pada geladak pertama .....	4 - 6
Gambar	4.6	Dialog untuk mrnampilkan gambar layout .....	4 - 6
Gambar	4.7	Gambar layout yang telah dipilih .....	4 - 7
Gambar	4.8	Tampilan atribut untuk <b>Toilet 13s</b> .....	4 - 8
Gambar	4.9	Gambar layout template yang telah dipilih, siap untuk diproses .....	4 - 9
Gambar	4.10	Pilihan proses <i>matching</i> pada suatu template .....	4 - 10
Gambar	4.11	Contoh hasil proses pilihan <b>TOPOLOGY</b> dengan hasil layout yang mempunyai tingkat kesamaan tertinggi secara topologi .....	4 - 11
Gambar	4.12	Contoh hasil proses pilihan <b>GEOMETRIC-SHAPE</b> dengan hasil tiga layout yang mempunyai tingkat kesamaan tertinggi secara geometrik dan topologi .....	4 - 12

## DAFTAR TABEL

		Halaman
Tabel	5.1. Hasil matching antara data layout Ship_Deck_4 dengan Template_Deck_4 .....	5 - 2
Tabel	5.2. Hasil matching antara data layout Ship_Deck_3 dengan Template_Deck_3 .....	5 - 3
Tabel	5.3. Hasil matching antara data layout Ship_Deck_2 dengan Template_Deck_2 .....	5 - 4
Tabel	5.4. Hasil matching antara data layout Ship_Deck_1 dengan Template_Deck_1 .....	5 - 5



# **BAB I**

## **PENDAHULUAN**





## BAB I

### PENDAHULUAN

#### 1.1. Latar Belakang

Salah satu bagian penting dalam perencanaan kapal adalah pembuatan Rencana Umum (*General Arrangement*). Inti dari pembuatan Rencana Umum adalah penentuan ruang akomodasi dan penempatannya pada geladak akomodasi. Dalam membuat sebuah desain tataletak seorang desainer akan terhubung kembali dengan pengalaman dan keahlian yang dimiliki sebelumnya (Foz, 1973; Akin, 1978; Jones, 1980). Hal ini berarti baik sengaja maupun tidak desainer akan menggunakan desain-desain lama yang telah ada (*design reuse*) untuk membuat desain baru.

Keuntungan jika desainer memanfaatkan desain-desain lama untuk pembuatan desain baru (Manfaat 1996) adalah :

1. Memperpendek proses desain

Dimana proses desain tidak dimulai dari awal karena menggunakan desain yang sudah ada sehingga mempercepat waktu perancangan.

2. Mengurangi banyak tugas-tugas yang diperlukan untuk pembuatan desain, karena desainer tidak merancang dari awal (*from scratch*).

3. Mencapai hasil desain yang lebih memenuhi permintaan pemesan (*owner*) atau persyaratan desain.

*Case Based Reasoning* (CBR) (Schank, 1982; Kolodner dkk., 1985; Hammond, 1989) merupakan pendekatan yang dapat mewujudkan



keuntungan di atas. Maher dkk. (1995) memperkenalkan suatu model *Case Based Reasoning* (CBR) in Design. CBR mempunyai sebuah representasi desain-desain lama (*case base*) dan dua proses, yaitu proses penarikan (*recall*) dan proses adaptasi desain lama (*adapt*).

Sebagai seorang manusia desainer mempunyai keterbatasan-keterbatasan. Salah satu keterbatasan tersebut adalah keterbatasan ingatan, yang mengakibatkan tidak semua desain terkoordinasi dengan baik. Disini komputer dapat digunakan sebagai *tools* untuk merealisasikan hal tersebut. Keuntungan yang didapat dengan pemanfaatan komputer untuk menunjang *desain reuse* adalah:

1. Membantu dalam menyimpan dan merepresentasikan desain lama.
2. Mendukung pemberian indeks pada desain.
3. Membantu dalam mengelompokkan desain.
4. Membantu pencarian (*searching*) dari desain lama yang sesuai dengan permasalahan desain (*new problem*).
5. Mendukung kesesuaian pola antara permasalahan desain dengan desain lama.
6. Membantu dalam pengukuran kesamaan antara *new problem* dengan desain lama.
7. Mendukung dalam pemilihan desain lama yang mempunyai kesamaan terbanyak dengan permasalahan desain.

Keuntungan di atas, sangat jelas bahwa pemanfaatan desain lama yang didukung komputer dapat mengurangi beban kerja desainer.



## 1.2. Perumusan Masalah

Berdasarkan uraian yang telah diuraikan diatas, pokok permasalahan yang harus dipecahkan adalah :

1. Bagaimana mengimplementasikan desain tata letak ruang akomodasi ke dalam tampilan komputer yang menarik dan mudah dimengerti oleh pemakai.
2. Bagaimana mempresentasikan desain tata letak ruang akomodasi yang sudah ada ke dalam case base.
3. Bagaimana menentukan kesamaan dari desain permintaan pemesan dengan desain-desain yang ada dalam case base.
4. Bagaimana memberi score derajat kesamaan desain dengan permintaan owner.
5. Bagaimana memilih salah satu (atau lebih) desain yang mempunyai score derajat kesamaan yang tertinggi.

## 1.3. Tujuan dan Manfaat

Penulisan tugas akhir ini bertujuan untuk memperbaiki proses desain tata letak ruang akomodasi dari desainer, sehingga di peroleh desain baru yang sesuai keinginan pemesan dalam waktu yang cepat. Tujuan utama ini dicapai melalui tujuan khusus di bawah ini :

1. Mengembangkan pendekatan *Case Based Reasoning* (CBR) dalam perancangan kapal.





2. Mengembangkan sebuah prototipe perangkat lunak komputer untuk mengimplementasikan pendekatan *Case Based Reasoning* (CBR) dalam merancang tata letak ruang akomodasi kapal.

Sedangkan manfaat yang diharapkan adalah :

1. Mempercepat proses perencanaan suatu kapal, karena perancangan desain tata letak ruang akomodasi membutuhkan waktu yang relatif lebih singkat.
2. Menghemat biaya produksi dalam pembuatan sebuah kapal.
3. Mengurangi tugas desainer, karena desainer tidak merancang dari awal.
4. Desain yang dihasilkan lebih mendekati dengan permintaan pemesan.

#### 1.4. Batasan Masalah

Pokok permasalahan akan lebih terarah jika permasalahan tersebut dibatasi dengan suatu batasan masalah. Dalam tugas akhir ini batasan masalah sebagai berikut :

1. Obyek permasalahan yang akan dipecahkan dibatasi untuk ruang akomodasi ABK kapal penumpang dengan kapasitas 1000 orang.
2. Pendekatan teknik *Case Based Reasoning* (CBR) untuk mendesain tata letak dibatasi sampai dengan proses penarikan (*recall*) desain dalam *case base*.
3. Teknik-teknik penarikan (*recall*) desain yang digunakan adalah teknik *geometric and topological pattern matching*, yaitu proses penarikan



desain dengan menunjuk pada kesamaan bentuk geometri dan hubungan antar ruang.

4. Data dalam *case base* merupakan data yang optimum.

### 1.5. Metodologi dan Model Analisis

Metode yang digunakan dalam penyusunan tugas akhir ini meliputi :

1. Studi Literature

Dilakukan studi literature tentang teori dan teknik dasar sebagai acuan dalam penulisan tugas akhir ini.

2. Pengumpulan Data

Pada tahap ini akan dikumpulkan data-data dari desain tata letak ruang akomodasi yang pernah ada sebagai data dalam *case base*.

3. Hipotesa Awal

Pada tahap ini dilakukan dugaan awal akan hasil dari data yang dikumpulkan terhadap hasil yang ingin dicapai.

4. Perumusan permasalahan

Pada tahap ini dilakukan perumusan permasalahan sebagai langkah-langkah dalam pemecahan masalah.

5. Pemecahan masalah

Pada tahap ini dilakukan pencarian solusi terhadap permasalahan-permasalahan yang telah dirumuskan sebelumnya dengan teori dan metode yang ada.

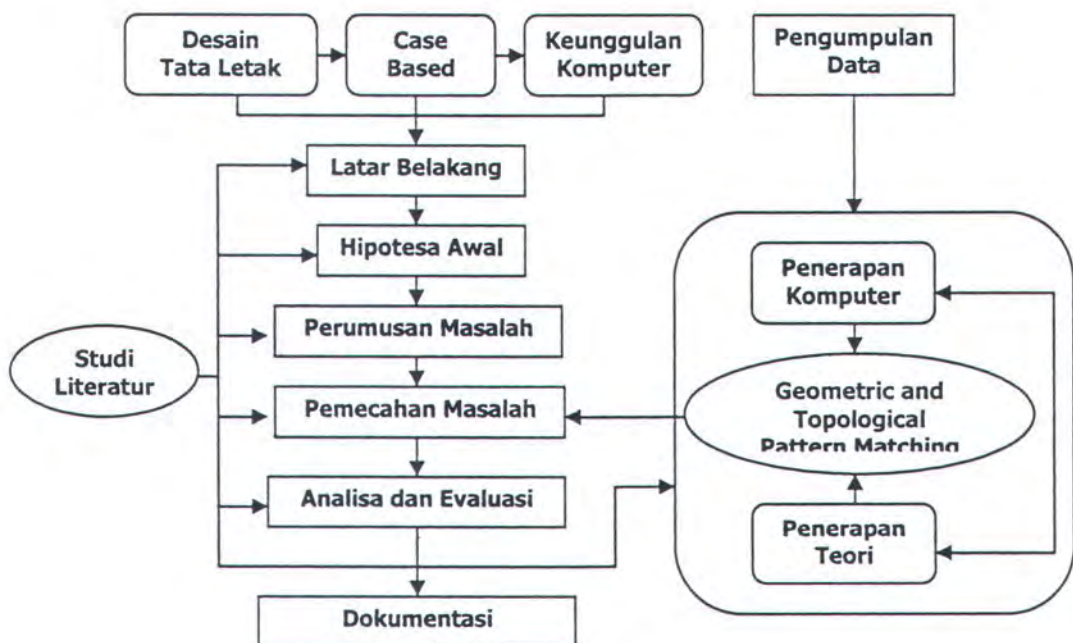


6. Analisa dan evaluasi

Hasil yang telah dicapai dievaluasi yang selanjutnya digunakan sebagai pertimbangan dalam memecahkan masalah desain tata letak ruang akomodasi kapal.

7. Dokumentasi

Tahap penulisan tugas akhir terhadap proses penelitian yang telah dilakukan di atas.



Gambar 1.1. Diagram metodologi dan analisis





## 1.6. Sistematika Laporan

### BAB I PENDAHULUAN

Bab ini mencakup latar belakang, tujuan dan manfaat, permasalahan, batasan permasalahan, metodologi dan model analisis serta sistematika laporan tugas akhir.

### BAB II TINJAUAN PUSTAKA

Bab ini berisi tinjauan pustaka mengenai konsep ruang akomodasi kapal dan konsep pembuatan desain tata letak ruang akomodasi kapal yang digunakan dalam penyelesaian masalah dalam tugas akhir ini.

### BAB III TEKNIK IMPLEMENTASI

Bab ini mencakup tahapan-tahapan proses perencanaan desain tata letak ruang akomodasi kapal, mulai dari data awal dan pengolahannya, penyamaan hubungan antar ruang akomodasi dan metode yang digunakan dalam memilih desain tata letak ruang akomodasi anak buah kapal yang sesuai dengan permintaan pemesan.

### BAB IV SISTEM SPACEMAN

Bab ini berisi penjelasan tentang penerapan teknik komputer yang digunakan dalam proses *pattern matching* dari layout ruang akomodasi anak buah kapal, yang diberi nama sistem SpaceMan.

### BAB V ANALISIS HASIL



Bab ini berisi bahasan tentang hasil analisa dari program yang telah dibuat, apakah hasil program tersebut sudah mencapai tujuan yang diinginkan atau belum.

## BAB VI KESIMPULAN DAN SARAN

Bab ini berisi kesimpulan dari penyelesaian masalah, tujuan yang ingin dicapai serta saran-saran terhadap upaya penyelesaian permasalahan selanjutnya.

## DAFTAR PUSTAKA

## LAMPIRAN





## BAB II

# TINJAUAN PUSTAKA







## BAB II

### TINJAUAN PUSTAKA

Dalam tinjauan pustaka ini akan dijelaskan beberapa landasan teori, aturan dan konsep yang berhubungan dengan aspek perencanaan ruang akomodasi kapal untuk anak buah kapal (ABK). Aspek yang dimaksud terutama berhubungan dengan proses desain secara umum yang diterapkan pada ruang akomodasi kapal, serta contoh model dan sistem yang telah dibuat oleh para pakar yang berkompeten.

#### 2.1. Rencana Umum

Rencana umum suatu kapal dapat didefinisikan sebagai pemberian ruang untuk semua fungsi dan peralatan yang disyaratkan, pengkoordinasian yang tepat untuk lokasi dan akses (Tapscott, 1980).

Menurut Tapscott (1980) ada empat langkah yang berurutan yang dilakukan dalam pembuatan rencana umum sebuah kapal, yaitu:

1. penyediaan ruang utama,
2. memberi pembatas pada ruang pribadi,
3. memilih dan menempatkan peralatan serta memperaboti dinding pembatas, dan
4. penyediaan akses yang saling menghubungkan antar ruang.

Langkah tersebut dimulai dari hal yang menyeluruh seperti penempatan ruangan-ruangan untuk ABK, tangki-tangki, ruang muat, dan sebagainya



sampai hal yang terperinci seperti perencanaan dari sebuah ruangan ABK, sirkulasi udara pada ruang muat dan sebagainya. Sebenarnya inti dari pembuatan Rencana Umum adalah penentuan ruang-ruang akomodasi dan penempatannya pada geladak akomodasi.

### **2.1.1. Ruang**

Menurut Kamus Besar Bahasa Indonesia (1990) ruang bisa diartikan dalam dua pengertian. Yang pertama, ruang adalah rongga yang terbatas atau terlingkup oleh bidang. Sedangkan yang kedua, ruang adalah rongga yang tidak terbatas, tempat segala yang ada. Secara harfiah “ruang” dapat diartikan sebagai alam semesta yang dibatasi oleh atmosfer dan tanah dimana kita berpijak, sedangkan secara sempit “ruang” berarti suatu kondisi yang dibatasi empat dinding, yang dapat diraba dan dirasakan keberadannya (Suptandar, 1999).

Penempatan bidang pembatas pada keempat sisi ruang bisa menimbulkan kesan bahwa ruang terasa sempit, luas, lebar, menyenangkan, menakutkan, formal dan sebagainya. Ruang adalah unsur penting dalam desain sebagai tempat kehidupan manusia dalam melakukan tugas dan kewajibannya. Ruang bagi manusia adalah segala-galanya sebagai tempat tinggal, sebagai status diri dan status sosial.

Secara garis besar ruang dibedakan atas (Suptandar, 1999):

1. Ruang nyata





Yaitu ruang yang dapat diukur secara nyata dan bisa dirasakan keberadaannya karena bentukan dari beberapa bidang atau komponen tertentu. Ruang nyata dibagi menjadi dua yaitu ruang terbuka dan ruang tertutup. Ruang yang berhubungan langsung dengan bagian luar disebut ruang terbuka.

## 2. Ruang abstrak

Yaitu ruang yang tidak ada batasannya dan tidak ada fakta yang nyata dan tidak mudah dipahami secara visual oleh setiap orang. Implikasi permasalahan membutuhkan kesadaran, kejelian atau kearifan manusia yang bersangkutan oleh karena ruang abstrak tidak nyata dan tidak diukur tetapi dapat dikhayalkan keberadaannya.

Menurut Goodban (1979) berdasarkan fungsinya, ruangan untuk hunian diklasifikasikan menjadi tiga golongan, yaitu:

### 1. Area pemukiman (*living area*)

Merupakan daerah yang digunakan sebagai ruangan-ruangan yang berfungsi untuk tempat bersosialisasi antara anggota keluarga dan untuk melakukan kegiatan-kegiatan yang bersifat rekreasi.

### 2. Area peristirahatan (*sleeping area*)

Merupakan daerah yang digunakan sebagai manusia beristirahat secara total, artinya waktu istirahat yang digunakan lebih lama. Yang termasuk dalam golongan ini adalah ruang tidur, kamar mandi dan penyimpanan pribadi.





### 3. Area pelayanan (*service/working area*)

Merupakan daerah yang berhubungan dengan penyediaan dan penyimpanan makanan. Selain itu area ini juga berfungsi sebagai tempat penyimpanan peralatan pemeliharaan rumah atau penyimpanan benda-benda yang bersifat umum. Contoh dari ruangan ini adalah dapur, baik dapur kering maupun dapur basah, gudang peralatan perawatan rumah dan gudang penyimpanan perlengkapan hobi.

#### 2.1.2. Ruang Akomodasi

Sebagaimana hunian, kapal merupakan sebuah struktur yang berhubungan dengan manusia sebagai pengoperasinya. Ruang akomodasi di kapal adalah ruangan-ruangan yang digunakan oleh ABK untuk tinggal di kapal selama kapal berlayar, dimana pada ruang akomodasi tersebut ABK saling berinteraksi dan berhubungan satu sama lain. Ruang akomodasi di kapal juga didefinisikan sebagai ruangan yang digunakan untuk ruangan publik, koridor, kamar mandi, kabin, kantor, klinik, ruang rekreasi dan lain-lain.

Ruang akomodasi ABK merupakan sebuah lingkungan dimana ABK saling berinteraksi dan berhubungan satu sama lain. Sehingga hal yang perlu diperhatikan dalam perencanaan ruang akomodasi kapal adalah keterkaitan antar individu, rekan kerja dan lingkungan tempat ia bekerja. Dengan demikian perencanaan ruang akomodasi di kapal juga perlu memperhatikan hal-hal yang berhubungan dengan



aspek psikologis sosial dimana lingkungan berpengaruh terhadap tingkah laku.

Enam aspek yang perlu diperhatikan dalam perencanaan ruang akomodasi kapal adalah (Smith dan Hatfield, 1977) :

1. Perlindungan dan keamanan (*shelter and security*)

Aspek ini merupakan aspek yang paling penting dan merupakan dasar kebutuhan dari tiap manusia. Dengan adanya keamanan maka seseorang akan merasa nyaman untuk beraktifitas.

2. Kontak sosial (*social contact*)

Perencanaan ruang akomodasi harus dapat memaksimalkan kontak antar individu agar terjadi interaksi dalam sebuah kelompok yang memiliki kesamaan minat dan kepentingan. Sehingga yang perlu diperhatikan adalah pemenuhan kebutuhan antara pribadi dan kontak social bagi individu.

3. Identifikasi simbolik (*symbolic identification*)

Dilakukan untuk menunjukkan perbedaan tingkat social. Hal ini dapat dilakukan dengan pengaturan tata letak, sehingga tiap tingkatan ABK memiliki daerah sendiri.

4. Alat bantu kerja (*task instrumentality*)

Dalam perencanaan, hal yang perlu diperhatikan adalah efektifitas dalam bekerja. Hal ini dapat dicapai dengan kemudahan akses ke tempat kerja, pencahayaan dan mechanical handling.



5. Keindahan dan kesenangan (*beauty and pleasure*)

Akomodasi di kapal hendaknya juga memiliki aspek keindahan.

Selain itu ruangan-ruangan di kapal dapat memenuhi kebutuhan ABK untuk berekreasi.

6. Pertumbuhan dan pengembangan (*growth and development*)

Pada aspek ini desain harus menciptakan aktifitas alternatif sehingga dapat memaksimalkan kontak social dan menyebabkan perkembangan antar individu.

Ruangan akomodasi kapal secara umum dapat dibagi dalam :

1. Ruang umum, adalah bagian dari ruang akomodasi kapal yang digunakan sebagai ruang makan, ruang duduk atau pada dasarnya adalah ruangan yang berfungsi sebagai tempat-tempat umum.
2. Lorong-lorong, yaitu yang digunakan untuk jalan-jalan yang menghubungkan satu tempat ke tempat lainnya.
3. Ruang saniter, yaitu ruangan yang digunakan untuk :
  - mandi (kamar berendam, kamar mandi dengan memakai shower),
  - kamar-kamar kecil (biasanya satu kamar dengan kamar mandi),
  - kamar penatu,
  - kamar cuci, dan
  - fasilitas-fasilitas saniter lainnya.





4. Kantor, merupakan ruangan yang digunakan khusus sebagai kelengkapan Administrasi baik yang berhubungan dengan kapal itu sendiri maupun yang berasal dari luar kapal.
5. Ruang private, yaitu ruang pribadi ABK yang biasanya terdiri dari ruang tidur dan ruang santai, atau ruang tidur dan ruang santai menjadi satu atau ruang tidur saja.
6. Ruang makan dan ruang persiapan makan. Ruangan ini biasanya harus ada pada setiap kapal dan ruangan ini terdiri dari :
  - *galley* (dapur), yaitu ruangan untuk keperluan masak,
  - *pantry*, yaitu ruangan yang digunakan untuk keperluan mempersiapkan makanan yang akan dihidangkan, dan
  - *messroom*, yaitu ruangan yang digunakan untuk makan-makan dan minum-minum.
7. Ruangan rekreasi dan olah raga, yaitu ruangan yang digunakan oleh para ABK untuk berolah raga atau untuk bersantai.

### 2.1.3. Ruang Akomodasi ABK

Ruang akomodasi dari ABK ditentukan oleh jumlah dan kebutuhan dari ABK, dan diusahakan didesain se nyaman mungkin. Karena sehari-harinya berada di atas laut sehingga ABK tidak cepat merasa bosan. Jumlah dari ABK ditentukan oleh kebutuhan pengoperasian kapal, tergantung dari besar dan tipe kapal, serta peralatan yang digunakan.



Ada perbedaan antara jumlah ABK pada kapal barang dan penumpang (Tapscott, 1980). Perbedaan tersebut adalah jumlah dari *steward's department crew*. Pada kapal penumpang mempunyai jumlah yang besar dibanding dengan kapal barang karena untuk melayani penumpang. Sebagai konsekuensinya jumlah ABK lebih banyak sehingga ruang akomodasi ABK terbatas.

Berdasarkan aturan dari *US Coast Guard Regulation* faktor yang menentukan besarnya ABK adalah

1. regulasi dan peraturan,
2. persyaratan dari owner untuk perawatan dan tugas steward, dan
3. gabungan persyaratan yang dihasilkan dari negosiasi antara perancang dengan owner.

#### 2.1.4. Tata Letak Ruang Akomodasi Kapal

Tata letak secara umum dapat didefinisikan sebagai pengaturan/peletakan beberapa subruang dalam satu ruangan (Manfaat, 1992). Dalam tata letak ruang akomodasi kapal, tata letak dapat didefinisikan sebagai pengaturan ruangan-ruangan akomodasi dalam suatu geladak akomodasi.

Banyak prosedur yang dibangun untuk merencanakan suatu tata letak. Dalam perkembangannya teori perencanaan tata letak saat ini, metode *Systematic Layout Planning* (SLP) yang dikembangkan oleh Muther (Francis, 1992) paling banyak digunakan.



Adapun persyaratan-persyaratan yang dipertimbangkan dalam perancangan tata letak ruang akomodasi ABK adalah :

1. Aliran pergerakan ABK

Dalam pembuatan tata letak ruang akomodasi kapal, aliran pergerakan ABK harus baik dan teratur agar tidak mengganggu aktifitas dari ABK.

2. Hubungan aktifitas ABK

Hubungan antar ABK dari segi aktifitas atau tugas yang harus dilakukan juga harus diperhatikan.

3. Penentuan kebutuhan dan ketersediaan ruangan

Kebutuhan minimum ruang akomodasi untuk semua ABK dapat ditentukan dengan memperhatikan ketentuan-ketentuan yang telah ditetapkan baik dalam ILO maupun Merchant Shipping Regulation. Selanjutnya dengan memperhatikan kebutuhan minimum ruang dan luas daerah total yang ada maka kita dapat mengalokasikan luas untuk masing-masing ruang.

4. Pertimbangan modifikasi

Sebuah kapal memiliki peluang untuk dikembangkan setelah jangka waktu tertentu mengikuti perkembangan dalam bidang perkapalan termasuk dalam tata letak ruang akomodasi. Untuk itu tata letak yang akan direncanakan diusahakan memiliki kemudahan dalam pemodifikasian.





## 5. Batasan-batasan

Dari peraturan yang digunakan (*Merchant Shipping Regulation* dan ILO), maka akan dapat dilihat batasan-batasan terhadap ruangan-ruangan maupun penempatan ruangan tersebut. Hal ini dijadikan pertimbangan dalam menyusun tata letak.

### 2.2. Pemanfaatan Desain Lama (*Design Reuse*)

Dalam membuat sebuah desain tata letak, seorang perancang akan terhubung kembali dengan pengalaman dan keahlian yang dimiliki sebelumnya (Foz, 1973; Akin, 1978; Jones, 1980). Hal ini berarti perancang baik secara sengaja atau tidak akan menggunakan desain-desain lama yang telah ada (*design reuse*) untuk membuat sebuah desain baru.

Menurut Manfaat (1996), pemanfaatan desain-desain lama untuk diproses menjadi desain baru mempunyai beberapa keuntungan :

1. Memperpendek proses desain

Dalam hal ini proses desain tidak dimulai dari awal, sehingga dapat mempercepat waktu perancangan.

2. Mengurangi tugas yang diperlukan untuk pembuatan desain, karena perancang tidak perlu merancang desain dari awal (*from scratch*).

3. Mencapai hasil desain yang lebih memenuhi permintaan pemesan (*owner*) atau persyaratan.

Model pendekatan yang dapat merealisasikan pemanfaatan desain lama adalah *Case Based Reasoning* (CBR) (Schank, 1982; Kolodner dkk., 1985;



Hammond, 1989). Dan diperkenalkan pula model pendekatan khusus desain oleh Maher dkk. (1995) yaitu *Case Base Reasoning in Design*.

### 2.3. Pemanfaatan Komputer untuk Menunjang *Design Reuse*

Dalam pemanfaatan desain lama yang telah ada (*design reuse*) perancang biasa menggunakan cara manual, yaitu dengan mengandalkan ingatan dari perancang. Hal ini mempunyai kekurangan tidak semua desain dapat terkoordinasi dengan baik karena adanya keterbatasan ingatan perancang. Disini komputer dapat digunakan sebagai *tools* untuk merealisasikan hal tersebut, yaitu dengan menggunakan memori komputer untuk menyimpan desain-desain lama dan perancang dapat memanggilnya setiap waktu.

Adapun pemanfaatan komputer untuk menunjang *design reuse* dapat diuraikan sebagai berikut :

1. Membantu dalam penyimpanan dan representasi desain lama.
2. Mendukung pemberian indeks pada desain.
3. Membantu dalam pengelompokan desain.
4. Membantu pencarian (*searching*) dari desain lama yang sesuai dengan permasalahan desain (*new problem*).
5. Mendukung kesesuaian pola antara permasalahan desain dengan desain lama.





6. Membantu dalam pengukuran kesamaan antara *new problem* dengan desain lama.
7. Mendukung dalam pemilihan desain lama yang mempunyai kesamaan terbanyak dengan permasalahan desain.

Dengan adanya dukungan komputer seperti diuraikan di atas, maka pemanfaatan komputer memberikan banyak keuntungan bagi perancang untuk menyelesaikan pembuatan desainnya. Perancang tidak perlu bingung dengan arsip desain yang dimilikinya, karena telah tersimpan dalam komputer dan jika suatu waktu dibutuhkan dapat ditampilkan secara cepat, rapi dan akurat. Selain itu komputer dapat membantu dalam pemilihan kesamaan antara desain input dengan desain lama yang tersedia. Program komputer (matematis, simulasi dan sebagainya) merupakan alat yang sangat berguna dan mempunyai kemampuan besar dalam membandingkan pilihan susunan suatu daerah kegiatan dalam batasan criteria yang terpilih dan data yang tersedia (Apple, 1997). Dari uraian diatas, sangat jelas bahwa pemanfaatan desain lama yang didukung oleh komputer dapat mengurangi beban kerja perancang.

Dalam tugas akhir ini penulis menggunakan program aplikasi (bahasa pemrograman) *Harlequin Lisp Work* (The Harlequin Group Limited, 1994) yang berbasis pada Windows. Program aplikasi ini mempunyai spesifikasi untuk penanganan data-data (*cases*) secara terstruktur (menjadi suatu *case based*) dan tiap-tiap statemennya mampu diakses untuk dieksekusi. Pemanggilan keseluruhan atau sebagian statemen dari suatu file yang berisi





listing program dapat dilakukan karena bentuk dari statemennya adalah fungsi (*function*).

#### 2.4. Case Based Reasoning

*Case Based Reasoning* (CBR) (Schank, 1982; Kolodner dkk., 1985; Hammond, 1989) merupakan suatu bentuk kecerdasan buatan (*artificial intelligence*) (Domeshek dan Kolodner, 1992). Ide dasar dari sistem *Case Base Reasoning* adalah bahwa secara umum seseorang mampu melakukan hal yang baik dalam memperhitungkan apa yang akan dilakukan dalam menghadapi masalah yang baru karena mereka mampu mengingat dan menyesuaikan hal-hal yang mereka lakukan (melihat atau lainnya) yang mirip dengan situasi sebelumnya (pengalaman masa lalu).

Sistem ini telah diterapkan pada penekanan *reasoning task* (Domeshek dan Kolodner, 1992) seperti *planning* (Hammond, 1989; Alterman, 1988), *scheduling* (Mark, 1989), pengambilan keputusan (*decision-making*) (Bain, 1986), *diagnosis* (Hunter dan Bareiss, 1989; Koton, 1988), *explanation* (Kass dkk., 1986; Ram, 1989), argumentasi (*argumentation*) (Ashley dkk, 1987; Simpson, 1985), dan *even design* (Hinrichs, 1991; Goel, 1989), CBR sedang dikembangkan dalam suatu teknologi untuk sistem pembangun (*building system*) yang membantu perancang dengan menunjukkan kepada mereka informasi yang sangat berguna yang dipilih dari ingatan pengalaman yang lalu yang terorganisasi



(*organized memories past experiences*) (Kolodner dkk., 1991; Domeshek, 1992).

Dalam sistem CBR, suatu permasalahan masa lalu dijadikan petunjuk untuk melakukan suatu tindakan atau menjadikan peringatan bagi sesuatu yang baru, yang mempunyai kontribusi untuk mengevaluasi atau menyesuaikan secara terperinci dari hal-hal yang mungkin terjadi (Domeshek dan Kolodner, 1992). Untuk mendapatkan suatu sistem yang menerapkan permasalahan dalam berbagai cara tersebut membutuhkan suatu pengetahuan istimewa yang sangat luas.

Cara kerja dari sistem CBR (Domeshek dan Kolodner, 1992) adalah perancang mendiskripsikan permasalahannya pada suatu sistem. Sistem tersebut menarik *case* yang mirip dan mempresentasikannya pada perancang. Perancang membuat keputusan tentang pemecahan permasalahan baru berdasarkan pada *case* tersebut, memperbarui permasalahan dan solusinya secara spesifik, dan mengulang proses tersebut. *Case* yang ditarik oleh sistem tersebut bisa menolong perancang dengan penjelasan saran untuk pemecahannya. Sistem tersebut cenderung kelihatan seperti sistem hipermedia yang cerdas (*clever hypermedia systems*); kecerdasan tersebut ditunjukkan oleh bagaimana informasi tersebut terorganisasi dan bagaimana hal itu dipresentasikan.



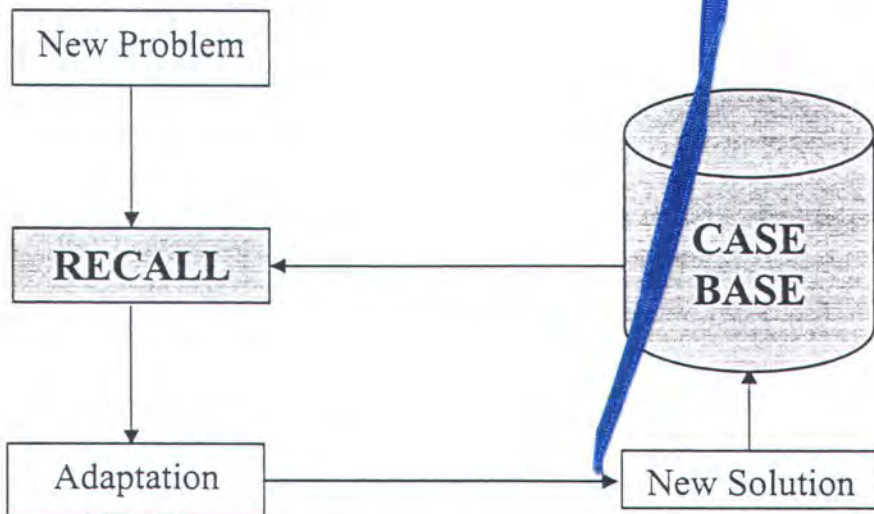


## 2.5. Case Based Reasoning in Design

Sistem *CBR in Design* dipertimbangkan sebagai suatu model desain yang difokuskan pada penggunaan kembali pengalaman desain (*design experience*) dalam bentuk tahapan (*episodes*), dari pada menyusun (*compiling*) atau menyamakan (*generalizing*) (Maher dan Zhang, 1991). *Episodes* tersebut mengacu pada keadaan permasalahan desain lama (*past design cases*) tertentu. Pada model ini analogi pemikiran (*analogical reasoning*) dipakai untuk memilih dan mentransformasi pemecahan desain lama tertentu menjadi solusi untuk suatu masalah desain baru (*new design problem*). *Analogical reasoning* mengacu pada pemikiran yang didasari gagasan bahwa permasalahan atau pengalaman di masa lalu bisa menyediakan beberapa pandangan atau acuan dalam memecahkan masalah baru (Maher dkk., 1995). Sebagai contoh, ketika mendesain interior dari geladak akomodasi suatu kapal, salah satu desainnya mungkin mengacu pada desain interior dari suatu rumah.

Maher dkk. (1995) mempersembahkan suatu model sederhana dari *CBR in Design* diilustrasikan pada gambar 2.1. Dalam model ini, *desain case* direpresentasikan dalam suatu *case base*. Diberikan suatu masalah desain baru, *case* akan ditarik dan kemudian diadaptasi untuk dijadikan pemecahan desain baru. Desain baru yang didapat bisa ditambahkan dalam *case base* sebagai sebuah *case* baru (*new case*).





Gambar 2.1. Model sederhana dari *Case Based Reasoning* (CBR) (Maher dkk. 1995)

Representasi dari *case* dalam *case base* menyediakan dasar untuk penggunaan kembali (*reuse*) *case* oleh sistem *CBR in Design* dan mendefinisikan bagaimana *reasoning* bisa diterapkan (Maher dkk., 1995). Dengan kata lain efisiensi dari proses penarikan *case* tergantung pada bagaimana *case* direpresentasikan dalam *case base*. Dalam proses penarikan *case*, terjadi pencarian *case* yang relevan dengan masalah desain baru. Setelah *case* yang relevan ditemukan, *case* tersebut diadaptasi. Pada proses adaptasi *case*, diputuskan mana yang akan diubah dan mana yang tidak (Maher dkk., 1995).

Dari uraian di atas model *CBR in Design* terdiri dari representasi desain-desain lama (*case base*) dan dua buah proses, yaitu proses penarikan (*recall*) dan proses modifikasi/adaptasi (*adapt*) desain lama yang telah ditarik. Secara singkat, dalam model ini jika diberikan masalah baru (*new problem*) sebagai input, maka desain lama akan ditarik dan selanjutnya akan



diadaptasi untuk menghasilkan desain baru (*new solution*). Desain baru tersebut kemudian dapat disimpan kedalam *case based* sebagai desain lama untuk siklus proses desain berikutnya. Representasi dari desain lama dalam case base disediakan sebagai dasar untuk penggunaan kembali desain lama dengan sistem CBR dan bagaimana pertimbangan dapat dilakukan. Dengan kata lain, efisiensi dari proses penarikan (*recall*) dari desain lama tergantung dari bagaimana desain-desain lama tersebut direpresentasikan dalam *case based*.

Langkah-langkah dari model *Case Based Reasoning in Design* adalah sebagai berikut :

1. Adanya permasalahan desain baru (*new problem*).

Permasalahan desain baru ini merupakan input dari *user* yang berupa gambar desain layout akomodasi ABK.

2. Perancang mengumpulkan desain-desain yang sudah ada (*desain reuse*) dalam *case base*.
3. Tahap penarikan salah satu desain di dalam *case base* (*recall*).

Dalam tahap ini dilakukan penarikan salah satu desain lama dalam *case base* yang sesuai dengan desain yang diinginkan *owner* (*new problem*). Sedangkan tahapan dalam *recall* adalah :

- a. *searching* (pencarian)

Merupakan salah satu proses pencarian desain-desain yang sesuai dengan keinginan *owner* (*user input*) dari dalam *case base*.





b. *matching* (mencari kesamaan)

Dalam tahap ini dilakukan pencarian kesamaan antara permasalahan desain (*new problem*) dengan desain yang diperoleh dari tahap *searching*.

c. *measuring similarities* (penilaian kesamaan).

Setelah didapatkan kesamaan-kesamaan dari permasalahan desain (*new problem*) dengan desain-desain lama (*design reuse*) yang dihasilkan dari tahap *searching*, selanjutnya dilakukan pengukuran/penilaian kesamaan, yaitu pemberian *score* dari desain-desain yang mempunyai kesamaan tersebut. Desain dengan *score* tertinggi merupakan desain yang mempunyai kesamaan terbanyak dengan permasalahan desain.

d. *retrieving* (penarikan)

Dalam tahap ini dilakukan proses penarikan desain-desain yang mempunyai *score* kesamaan tertinggi yang dihasilkan dari tahap *measuring similarities*.

e. *selecting* (pemilihan)

Proses yang terakhir dari tahap *recall* ini merupakan pemilihan desain (*selection*), dimana dalam hal ini perancang tidak harus memilih desain yang mempunyai *score* kesamaan tertinggi yang dihasilkan dari proses *retrieving*.





4. *Adaptation*, yaitu tahap memodifikasi desain yang sudah dihasilkan dari tahap *recall*.

Pada proses adaptasi ini, dibuat keputusan bagian mana dari desain lama yang perlu dirubah dan bagian mana yang dibiarkan tetap sama (Maher dkk., 1995).

5. Didapatkan *new solution* yang sesuai dengan keinginan pemesan (*owner*).
6. *New solution* ini dapat dipakai/dimasukkan dalam *case base* sebagai referensi untuk siklus proses berikutnya.

#### 2.5.1. Representasi *Design Case*

Menurut Maher dkk. (1995), pengembangan *case memory*, atau suatu *case base* meliputi definisi:

- Isi dari sebuah *design case*.
- Representasi dari isi sebuah *design case*.
- Organisasi dari kumpulan *design case* dalam *case memory*.

##### 2.5.1.1. Isi dari *design case*

*Design case* mempunyai beberapa bentuk yang berbeda, diantaranya yaitu sejarah desain, persyaratan dan penyelesaian desain, perbedaan pendapat dalam permasalahan desain dan proses penyelesaiannya (Maher dan Garza, 1997).

Sejarah desain (*design stories*) dapat digunakan perancang untuk mempelajari situasi/kondisi yaitu kondisi baik dan buruk dari



penyelesaian desain-desain lama yang telah ada, dan pengambilan keputusan dalam penyelesaian *design case* lama tersebut. Selain itu sejarah desain dapat digunakan untuk membantu perancang dalam mengambil keputusan ketika memecahkan masalah desain yang baru, atau masalah desain yang hampir sama.

Persyaratan desain lama yang berkaitan dengan *design case* dapat digunakan untuk mendaftar ciri-ciri yang digunakan dalam menyelesaikan kasus yang ada. Hal tersebut dapat langsung dibandingkan dengan persyaratan desain baru untuk mendefinisikan gambaran umumnya. Sebagai tambahan, dapat diambil bentuk dari batasan desain yang dihubungkan dengan permasalahan desain. Penyelesaian desain lama (dalam bentuk penggambaran desain) dapat digunakan sebagai penyelesaian awal untuk masalah desain baru (*new design problem*). Untuk penyelesaian kembali *design case* baru, penyelesaian desain lama disusun secara khusus dalam sebuah file yang disimpan dalam *case base* (Maher dkk., 1995).

Perbedaan pendapat dari masalah desain mengacu pada perbedaan representasi dari desain lama. Ada dua teori yang mendukung hal tersebut, yaitu : teori kemampuan (*accessibility theory*) yang merepresentasikan hubungan topologi antar ruang pada susunan tata letak berdasarkan kedekatannya, dan teori *night zone distribution* yang merepresentasikan hubungan pertukaran antar ruangan dalam sebuah layout.





Dalam sistem Dave dkk., permasalahan desain direpresentasikan dalam gambar geometri yang dilengkapi dengan informasi simbolik dan hubungan topologi antar ruang dalam tata letak yang didasarkan pada hubungan kedekatannya.

Proses penyelesaian yang digunakan dalam menyelesaikan permasalahan desain dapat digunakan kembali untuk menyelesaikan permasalahan desain yang baru.

#### 2.5.1.2. Representasi dari isi *design case*

Representasi dari isi *design case* mengacu pada bentuk dimana isi dari *design case* disimpan dalam *case base*. Kemungkinan akan didapat bentuk yang berbeda, bentuk yang paling umum adalah sepasang nilai atribut (*attribute-value pairs*). (Kolodner, 1993; Giretti dkk., 1994; Vop dkk., 1994; Maher dkk., 1995; Flemming dkk., 1997). Bentuk yang lain meliputi *text* (Domeshek dan Kolodner, 1992; Kolodner, 1993; Gross dkk., 1994), persamaan dan pertidaksamaan matematika (*mathematical equalities and inequalities*) (Hua dkk., 1996; Faltings, 1997), penggambaran berdasarkan grafik (Giretti dkk., 1994; Vop dkk.; Dave dkk., 1994), *symbol/icon* (Giretti dkk., 1994; Vop dkk., 1994) dan representasi multimedia (Domeshek dan Kolodner, 1992; Gross dkk., 1994).

*Design case* yang direpresentasikan dengan sepasang nilai atribut dijelaskan dengan atribut-atribut dimana masing-masing atribut memiliki nilai sendiri. Sebagai contoh sebuah bangunan





gedung dapat digambarkan dengan informasi geometriknya (seperti panjang, lebar, luas dan lain-lain) sebagai atributnya, dimana masing-masing mempunyai nilai sendiri.

*Text* dapat digunakan untuk menggambarkan sejarah desain (Kolodner, 1993) yang menjelaskan arsitektur *design case*. Sistem ARCHIE dan Gross dkk. menggunakan representasi ini.

Persamaan dan pertidaksamaan matematika digunakan untuk menggambarkan batasan desain yang berada dalam kotak desain. Sebagai contoh, dalam CADRE dan FAMING, persamaan dan pertidaksamaan matematika digunakan untuk menggambarkan satu set batasan menurut angka dalam dimensi rencana dasar *design case*.

Grafik dapat digunakan untuk menggambarkan hubungan atribut-atribut atau bagian dari *design case*, dimana *node* menggambarkan sebuah atribut atau bagian dari busur hubungan atau ketergantungan antara dua nodes. Sistem ASA, FABEL dan Dave dkk. menggunakan grafik untuk menggambarkan topologi (hubungan kedekatan antar ruang) dari sebuah layout.

*Symbol/icon* digunakan untuk memberikan gambaran khusus dari layout, seperti yang digunakan dalam ASA dan FABEL. Dalam ASA, simbol digunakan untuk menggambarkan teori *night zone distribution*.

Representasi multimedia merupakan kombinasi dari sejumlah *design case*. Sistem Gross dkk. menggunakan bentuk ini untuk



mempermudah pemahaman terhadap permasalahan dengan memberikan beberapa model pemahaman.

#### 2.5.1.3. Pengorganisasian *design case* dalam memori *case base*

*Design case* dapat diatur dalam memori *case base* sebagai suatu kumpulan, dimana perancang dapat meninjau kembali dan menyeleksi setiap permasalahan secara langsung. Sistem CADRE, FAMING dan Dave dkk. menggunakan pengaturan memori *case base* jenis ini. Fokus utama dari sistem ini adalah adaptasi permasalahan, menyeleksi sisa-sisa *design case* untuk perancang. Pengumpulan *design case* dalam memori *case base* dapat disusun sesuai rencana pengurutan (*indexing scheme*) (Maher dkk., 1995). Dalam pengorganisasian jenis ini, masing-masing *design case* diindeks sendiri-sendiri dan disimpan dalam *case base*.

Dalam sistem ARCHIE, masing-masing desain berisi sejarah desain. Tiap sejarah desain diurutkan dengan sekumpulan kombinasi ciri-ciri desain, seperti persoalan pokok desain (seperti keselamatan, kebebasan, kegaduhan dan kemampuan), komponen struktural (seperti lantai, ruangan dan bangunan samping) dan sistem fungsional (seperti sirkulasi, perpipaan dan listrik). Dalam hal ini, sejarah desain dari salah satu permasalahan desain saling terhubung satu dengan lainnya, dengan demikian hubungan antar permasalahan desain memungkinkan perancang untuk melihat dari satu permasalahan ke permasalahan lainnya.





### 2.5.2. Penarikan *Design Case*

Penarikan *design case* mengacu pada pencarian permasalahan-permasalahan dimana ciri-cirinya mempunyai kesamaan atau bersangkutan dengan ciri-ciri dari permasalahan desain baru. Hal tersebut dihasilkan dengan mendapatkan kembali permasalahan menggunakan metode penarikan permasalahan.

Metode ini dapat dikelompokkan dalam dua kategori, yaitu:

- *User-directed case selection* (langsung ditentukan oleh perancang)
- *Pattern matching-based layout case retrieval* (berdasarkan pada proses *pattern matching*).

#### 2.5.2.1. *User-directed case selection*

Dalam kategori ini perancang (*user*) mempunyai kendali dalam memilih *design case* dari suatu *case base*. Mereka bisa mencari dari *case base* tersebut dan memilih sebuah *design case* yang sesuai dengan keinginan mereka dimana *case* yang diambil relevan dengan masalah desain yang mereka.

Sistem ARCHIE, FAMING dan Dave dkk. menggunakan metode penarikan desain yang sesuai dengan kategori ini. Selanjutnya, untuk membantu perancang (*user*) menyelesaikan *design case* menjadi lebih efisien, ARCHIE menggunakan metode ini berdasarkan pada urutan indeks, dimana sejarah desain diurutkan dengan menggunakan gambaran/penjelasan desain yang ada.







#### 2.5.2.2. *Pattern matching-based layout case retrieval*

Kategori kedua ini meliputi proses pencarian suatu *case*, mencocokkan (*matching*) *case* tersebut, dan menentukan/menilai atau mendefinisikan kemiripan antara permasalahan desain baru dengan *case* tersebut, dan memilih *case* yang relevan. Dalam kategori ini *pattern matching* dipertimbangkan menjadi suatu proses kunci (Maher dkk., 1995). Pola (*pattern*) yang mencirikan masalah desain baru dicocokkan dengan pola yang mencirikan *case*.

Sistem Gross dkk. adalah contoh sistem yang sesuai dengan kategori ini, dimana pola penyamaannya menggunakan simbol (*symbolic pattern matching*).

#### 2.5.3. *Case Adaptation*

Adaptasi *design case* dititikberatkan pada modifikasi pilihan desain yang didapatkan dari *case base* untuk mendapatkan penyelesaian desain baru yang memenuhi persyaratan desain. Ada beberapa metode adaptasi kasus yang berbeda yang digunakan dalam sistem penggambaran yang dapat dikategorikan menjadi metode informal dan formal.

Metode informal mengacu pada metode yang dalam permasalahan proses adaptasi ini dibuat langsung oleh perancang (user). ARCHIE dan Gross dkk. menggunakan metode yang termasuk dalam kategori ini, tetapi fokus dari sistem-sistem tersebut adalah untuk menyajikannya sebagai suatu penjelajah kasus (*browser case*)



yang menekankan penyampaian informasi tentang design case kepada para perancang (user) yang lebih baik daripada adaptasi design case yang diberikan kepada perancang (user) secara langsung.

Metode formal mengacu pada metode yang menggunakan algoritma- algoritma untuk menunjang adaptasi desain kasus. Metode ini terdiri dari batasan pemenuhan persyaratan, teknik pengubahan topologi, pergerakan dan pengujian, perencanaan ulang desain.

Metode batasan pemenuhan persyaratan diterapkan untuk adaptasi kasus yang digambarkan sebagai permasalahan batasan pemenuhan persyaratan (Maher dkk., 1995; Hua dkk., 1996). Karena itu adaptasi design case bertujuan untuk memenuhi seperangkat batasan yang diinginkan dari permasalahan desain yang baru. Jenis-jenis adaptasi kasus dimana diterapkan metode batasan pemenuhan persyaratan meliputi adaptasi dimensional dan adaptasi desain obyek dari permasalahan desain layout. Adaptasi dimensional mengacu pada modifikasi dimensi design case layout. Sistem Dave dkk. menggunakan metode batasan pemenuhan persyaratan untuk adaptasi dimensional dari design case layout. FABEL menggunakan metode batasan pemenuhan persyaratan untuk menyelesaikan permasalahan penempatan desain obyek dalam design case layout yang diadaptasi.

Teknik-teknik perubahan topologi digunakan untuk adaptasi topologi, yaitu modifikasi dari hubungan topologi antar elemen design case layout, seperti contoh : kedekatan antar ruang.





Metode mencoba dan menghasilkan digunakan dalam FABEL. Dalam metode ini diberikan susunan layout, penarikan kembali design case layout dievaluasi untuk mengetahui apakah penempatan obyek dalam susunan layout bertentangan dengan elemen design case atau tidak. Apabila terjadi perbedaan, penempatan dari obyek desain dimodifikasi sesuai dengan permasalahan desain. Modifikasi dari susunan layout ini kemudian digunakan untuk menyelesaikan kembali design case yang lain. Keseluruhan proses ini diulang sampai penyelesaian layout yang sesuai dengan persyaratan didapatkan. Operasi pengulangan tersebut memungkinkan perancang untuk menentukan penambahan, pemindahan, pengumpulan obyek desain dan atributnya.

## 2.6. Pendekatan *Pattern Matching*

*Pattern matching* merupakan inti dari *Case Based Reasoning* (CBR), dimana *pattern matching* ini digunakan untuk menentukan tingkat kesamaan hubungan antara input dari perancang dengan data-data desain yang ada dalam case base.

### 2.6.1. Teknik-teknik *pattern matching*

Pendekatan *pattern matching* difokuskan pada empat teknik yang berbeda, yaitu: *geometric pattern*, *topological relations*, *distributed pattern* dan *semantic/symbolic pattern* (Manfaat dkk., 1996). Keempat teknik ini menunjukkan adanya perbedaan dasar





dalam representasi *pattern* yang digunakan dalam *teknik pattern matching*.

### 1. *Geometric pattern*

Dalam teknik ini *pattern* didefinisikan dalam sistem koordinat, dimana representasi dari teknik ini adalah kumpulan titik, *dimensional graph* dan gambar.

Kumpulan titik didefinisikan sebagai kumpulan dari sejumlah koordinat dalam sistem koordinasi dua dimensi. *Dimensional graph* menunjukkan *graph* yang memiliki elemen-elemen *vertices*, *edges* dan *faces* yang didefinisikan dalam sistem koordinat. *Graph-graph* tersebut dapat berbentuk dua dimensi (*planar graphs*) atau tiga dimensi (*polyhendra*). Sedangkan gambar dalam *geometric pattern* terdiri dari bermacam-macam bentuk, yaitu : *lines*, *arc*, *circles*, *polygons*, *curves* dan *characters*.

### 2. *Topological relations*

Teknik ini menggambarkan *pattern* sebagai bentuk hubungan antara elemen-elemen dari sebuah *pattern*. *Topological relations* ini direpresentasikan dalam bentuk *graph* yang tersusun atas *vertices*, *edges* dan *faces*. *Vertices* digambarkan sebagai elemen dari *pattern*, *edges* digambarkan sebagai hubungan antar elemen dan *faces* merupakan kumpulan kedekatan dari *edges*.

### 3. *Distributed pattern*



Teknik *distributed pattern* merupakan teknik yang mendistribusikan pattern ke dalam bentuk matriks (*grid units* atau *pixels*). Contoh representasi teknik ini adalah *spatial patterns*, *spatial relationship patterns*, dan *graphical patterns*.

*Spatial patterns* menghubungkan pola-pola yang memiliki bentuk shapes dari faces. Masing-masing grid unit atau pixels dari pattern merepresentasikan bagian shape dari space.

*Spatial relationship patterns* menghubungkan pola-pola yang merepresentasikan hubungan antara spaces pada spatial layout, dimana beberapa grid unit atau pixel merepresentasikan hubungan kedekatan antara dua ruang dan merepresentasikan posisi dari ruang tertentu yang memiliki kedekatan.

*Graphical patterns* ini pada dasarnya merepresentasikan bagian dari pola, tetapi tidak dalam bentuk spatial pattern.

#### 4. *Semantic/symbolic pattern*

Pada *semantic/symbolic pattern* dilakukan representasi atribut-atribut *semantic/symbolic* dari obyek yang meliputi nama, fungsi, ukuran, warna dan atribut lainnya. Atribut-atribut tersebut dapat direpresentasikan dalam bentuk teks atau simbol (seperti dalam diagram, icon dan lainnya). Pasangan nilai-nilai atribut dapat direpresentasikan dalam bentuk daftar (list).

Sistem dari Dave dkk. (1994), Giretti dkk. (1994), dan Gross dkk. (1994) yang berdasarkan pada CBR menggunakan





*symbolic pattern* untuk melakukan *pattern matching* sebagai bagian dari proses penarikan desain (*recall*).

Sistem Giretti, dkk. menggunakan *symbolic pattern* untuk menunjang representasi *planar graph* yang merepresentasikan *spatial relationship* dari *spatial layout*. Sedangkan pada sistem Gross dkk., *symbolic pattern* digunakan dalam bentuk diagram dan dihubungkan dengan program komputer untuk menginterpretasikan sketsa gambar tangan dan diagram. Sketsa ini dibentuk dari kumpulan primitif elemen yang disebut *glyphs*. Contoh dari *glyphs* adalah *lines*, *circles*, *boxes*, *triangles*, *arrows*, *numbers*, *letters* dan lain-lain.

Dalam Tugas Akhir ini penulis menggunakan teknik kombinasi antara *topological relations* dengan *geometric pattern*. Dan teknik ini bernama teknik *geometric and topological pattern matching* untuk melakukan proses penarikan (*recall*) desain lama. Teknik ini digunakan dengan alasan mudah dipahami oleh pengguna karena secara visual dapat dilihat bentuk dari ruangan yang di-*match*.

#### 2.6.2. Metode *matching*

Metode *matching* merupakan metode yang digunakan dalam proses *pattern matching* yang bertujuan untuk menentukan tingkat kesamaan antara pola dari input perancang (*user*) dengan desain-desain yang ada didalam *case base*. Ada tiga metode yang dipakai





dalam pattern matching, yaitu: *neural networks*, *mathematical technique* dan *symbolic pattern matching*.

### 1. *Neural networks*

Metode ini merupakan model matematis dari struktur otak manusia (Coyne dan Newton, 1989), yang disebut juga *parallel distributed processing system* (Rumelhart, 1989). Metode ini terdiri dari sejumlah jaringan unit proses (*nodes*) yang berisi data dan dihubungkan dengan garis penghubung (*weighted arcs*). Dalam metode ini input akan dimatching dengan sejumlah data yang sudah ada dan dengan proses iterasi akan diperoleh kesamaan antara input dengan data.

Metode *neural networks* ini mampu mengenali pola-pola yang tidak lengkap atau tidak teratur dan menghasilkan pola baru dengan pola yang lengkap. Kemampuan ini sangat mendukung dalam pengenalan desain, kontrol proses dan lainnya.

### 2. *Mathematical technique*

Metode *mathematical technique* ini banyak digunakan dalam permasalahan yang mengandung *geometric* dan *graph matching*. Pada intinya metode ini menggunakan teori matematika untuk menganalisa kesamaan antara pola-pola yang berbeda. Beberapa metode yang digunakan dalam matematika teknik yaitu: *coordinat distance measurement*, *hashing method* (Akutsu, 1995), *dynamic method* (Akutsu, 1995), *cross correlation*



(Cooper, 1989) dan *graph theoretic algorithms* (Hopcroft dan Trajan, 1973).

3. *Coordinate distance measurement*

Metode ini digunakan untuk mengukur jarak antara titik, garis, kurva dan lainnya dalam sistem koordinat. Schneider dkk. (1989), menggunakan metode ini untuk mengukur kesamaan bentuk antara dua gambar .

4. *Hashing method*

Metode ini memberikan akses langsung dan cepat pada data tertentu dengan cara memberi nilai tertentu dalam data base. Digunakan dalam proses matching bentuk tiga dimensi yang merupakan kombinasi antara *dynamic programming technique* dengan *Kabsch's method* (1976). Tujuannya adalah meningkatkan efisiensi proses pattern matching.

5. *Cross correlation*

Berdasarkan pada *method template matching* yang mengukur kesamaan antara template dengan data/gambar .

6. *Graph theoretic algorithms*

Metode ini berdasarkan pada algoritma untuk mencari kesamaan antara graph. Ada beberapa tipe kesamaan antar graph (Harary, 1969; Berger, 1976) yaitu:

- *Graph isomorphism*, mencari kesamaan antara *graph* dengan *graph* lainnya.



- *Subgraph isomorphism*, mencari kesamaan antara *graph* dengan *subgraph* dari *graph* lainnya.
- *Double subgraph isomorphism*, mencari kesamaan antara *subgraph* sebuah *graph* dengan *subgraph* dari *graph* lainnya.

Ballard dan Brown (1982), memberikan contoh dari ketiga tipe diatas yaitu *matching metric*, *backtrack search* dan *associated graph matching* (pembentukan struktur data dari dua *graph* yang dimatching).

#### 7. *Symbolic pattern matching*

Dipakai untuk proses matching dimana obyek mempunyai pola berupa simbol-simbol. Dalam prosesnya, pola-pola itu akan direpresentasikan sebagai simbol.

### 2.7. Sistem SPIDA



Sebuah sistem komputer yang bernama SPIDA (*SPatial Intelligent Design Assistant*) yang mempunyai kemampuan melakukan *topological pattern matching* dan kombinasi *topological pattern* dan *geometric shape matching* untuk *spatial layout* secara umum telah dikembangkan oleh Manfaat (1998).

*Topological pattern matching* antar layout bertujuan untuk menentukan kesamaan *topological pattern* diantara layout-layout tersebut. Dalam proses *pattern matching* jenis ini , *association graph technique*





(Ballard dan Brown, 1982) digunakan. Dengan ini SPIDA mampu menemukan jumlah maksimum ruangan yang fungsi-fungsinya bersesuaian diantara dua layout yang dibandingkan (*di-pattern matching*). Kemudian SPIDA mampu mengurutkan tingkat kesamaan antar layout yang dapat dipilih oleh desainer untuk digunakan sebagai desain awal dalam proses desain baru.

Seperti halnya *topological pattern matching* dalam kombinasi *topological pattern* dan *geometric shape matching*, SPIDA mampu menentukan tidak hanya kesamaan topologinya tetapi sekaligus juga kesamaan bentuk antar ruangan-ruangan yang fungsinya bersesuaian yang dihasilkan dari proses *topological pattern matching*. Selain *association graph technique* yang digunakan untuk mengukur kesamaan *topological pattern technique*, *planar shape matching* yang dikembangkan oleh Leu dan Huang (1988) digunakan untuk mengukur bentuk antar ruangan. Teknik yang terakhir ini dapat memberikan hasil yang akurat dimana pengukuran kesamaannya ditentukan berdasarkan *dissimilarity scores*, artinya tingkat perbedaan bentuk antar dua ruangan adalah yang diukur. Dengan demikian semakin besar *score* ini maka semakin berbeda kedua ruang tersebut. Atau dengan kata lain semakin kecil *score* tersebut maka tingkat kesamaan bentuk ruang tersebut semakin besar.



## **BAB III**

# **TEKNIK IMPLEMENTASI**



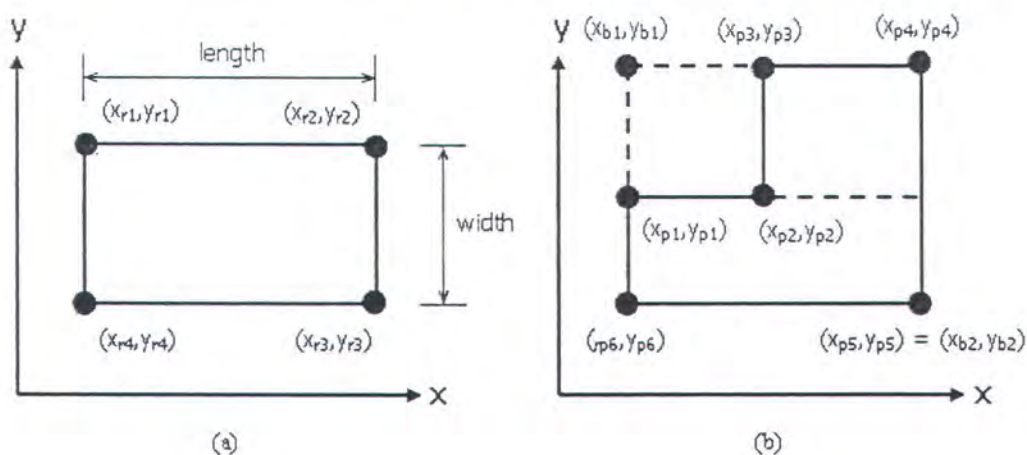


## BAB III

### TEKNIK IMPLEMENTASI

#### 3.1. Representasi Ruang

Dalam tugas akhir ini, ruang didefinisikan sebagai bangun yang merupakan persegi panjang (*rectangle*) (termasuk bujur sangkar) atau poligon (*polygon*) dua dimensi. Ruang tersebut direpresentasikan dalam suatu sistem koordinat cartesian dua dimensi, seperti gambar berikut:



Gambar 3.1. Grepresentasi ruang bentuk *rectangle* (a), dan *polygon* (b) (Manfaat 1998).

Untuk merepresentasikannya, ruang tersebut harus mempunyai beberapa atribut, yaitu:

- *space name*

Nama dari ruang sebagai identitas ruang tersebut, contoh *boys2\_0102s*.

Menunjukkan bahwa ruangan tersebut adalah ruang ke-2 untuk boys, dimana letaknya adalah pada *deck* 1 untuk kapal (*ship*) ke-2 yang





ditunjukkan oleh huruf *s*, untuk membedakan antara data pada *case based* dengan ruangan sebagai *template*. Sedangkan pada *template* digunakan huruf *t* pada akhir penamaan.

Data *template* adalah suatu data contoh yang direncanakan oleh desainer maupun permintaan owner.

- *space caption*

Memberi keterangan yang menunjukkan identitas ruang tersebut. Contoh *boys2 12s*. Digunakan untuk memberi keterangan suatu ruangan pada saat direpresentasikan dalam bentuk gambar.

- *space function*

Menunjukkan fungsi dari ruangan, apakah ruangan tersebut digunakan untuk ruang pribadi, ruang istirahat, kantor, dapur dan sebagainya.

Contoh pada ruang yang mempunyai nama *boys2\_0102s*, mempunyai fungsi *boys\_room* yaitu ruangan pribadi untuk boys.

- *space identifier*

Data ruangan tersebut terbagi menjadi dua bagian, yaitu satu sebagai data pada *case based*, sedangkan bagian yang lain sebagai data *template*. Untuk mengelompokkannya digunakan *identifier*. Ada dua macam *identifier* yaitu *Stored\_Object* untuk menunjukkan data *case based* dan *Template\_Object* untuk menunjukkan data *template*.

- *space shape*

Bentuk dari ruangan ditunjukkan pada atribut ini. Ada dua macam bentuk yaitu *rectangle* (persegi panjang) dan *base polygon* (poligon



dasar). Poligon dasar ini terdiri dari komponen persegi panjang (*rectangle*) dan bujur sangkar (*square*). Dan ditunjukkan pula koordinat *vertices*-nya dan masing-masing komponen penyusunnya.

- *space bounding*

Menunjukkan koordinat dari *bounding* (suatu persegi panjang yang membatasi ruangan tersebut) ruangan tersebut.

- *tree rep*

Merupakan bentuk representasi *binary tree* dari ruangan. Digunakan untuk proses *shape matching*.

- *poly points*

Berisi data titik yang merupakan *outline coordinates* yang digunakan untuk menentukan *binary tree*.

- *space area*

Luas dari ruangan tersebut ditunjukkan oleh atribut *area\_ruang*.

- *space color*

Menunjukkan atribut warna dari ruangan tersebut pada saat direpresentasikan dalam gambar.

### 3.2. Representasi Layout

Seperti pada ruang layout mempunyai beberapa atribut yang menggambarkan *properties* dari layout tersebut.

Atribut tersebut yaitu:



- *layout name*

Menunjukkan nama dari layout, dimana nama tersebut menunjukkan jenis, urutan dan letak layout tersebut.

Contoh *Ship\_03\_Deck\_02*, *Ship* menunjukkan jenis data, apakah termasuk data *case based* atau data *template*. Angka *03* setelah *Ship* menunjukkan urutan data dan Angka *02* setelah *Deck* menunjukkan letak layout tersebut di geladak kedua.

- *layout spaces*

Layout terdiri dari beberapa ruang, *spaces\_layout* menunjukkan ruangan-ruangan yang termasuk dalam suatu layout.

- *layout assignment*

Seperti yang ditunjukkan *name\_layout*, layout-layout tersebut dibedakan atas jenis data dan letaknya. Atas jenis data dibagi menjadi dua yaitu *case base* dan *template*, sedangkan berdasar letaknya dibagi menjadi 4 deck, sehingga ada 8 macam jenis layout. Jadi atribut ini menunjukkan termasuk jenis yang mana layout tersebut.

- *layout closeness*

Menunjukkan topologi (kedekatan antar ruang) dalam layout tersebut.

### 3.3. Topological Pattern Matching

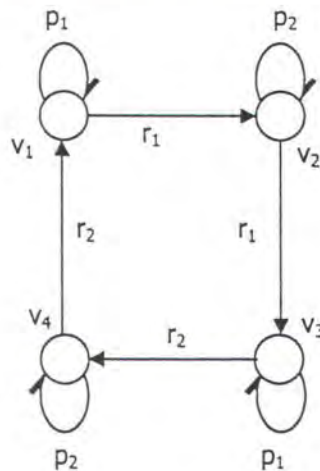
Topologi dari suatu layout direpresentasikan dalam suatu *graph*.

*Graph* tersebut terdiri atas beberapa kumpulan *node V*, kumpulan *property P* dari *node* tersebut dan kumpulan *relation R* yang mendefinisikan





pasangan dari *node*. Suatu contoh dari suatu representasi *graph* diberikan pada gambar 3.2. *Node*  $v_1$  dan  $v_3$  mempunyai *property*  $p_1$ ,  $v_2$  dan  $v_4$  mempunyai *property*  $p_2$ , dan egde antara  $v_1$  dan  $v_2$  dan antara  $v_2$  dan  $v_3$  masing-masing mempunyai *relation*  $r_1$ , dan antara  $v_3$  dan  $v_4$  dan antara  $v_4$  dan  $v_1$  masing-masing mempunyai *relation*  $r_2$ .



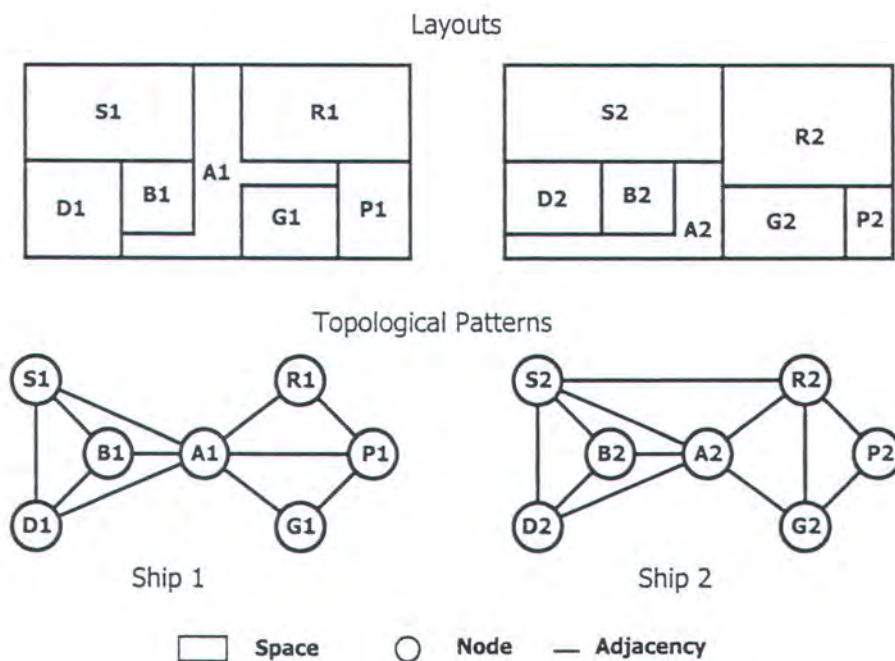
Gambar 3.2. Representasi sebuah *graph* (Ballard dan Brown, 1982)

Diberikan dua *graph* didefinisikan dengan  $(V_1, P, R)$  dan  $(V_2, R, P)$ , untuk mencocokkan dua *graph* tersebut sebuah *association graph* digambar. Suatu *association graph* adalah suatu *graph* yang *node*-nya adalah pasangan *node* dari dua *graph*, yang mempunyai kemiripan *properties*, dan *egde* menghubungkan *node* dari *graph* tersebut yang menunjukkan keadaan yang *compatible*. Untuk menggambar  $G$ , gambar sebuah *node* dari  $G$  dengan label  $(v_1, v_2)$  untuk setiap  $v_1$  dalam  $V_1$  dan  $v_2$  dalam  $V_2$  dengan syarat bahwa  $v_1$  dan  $v_2$  harus mempunyai kesamaan *property*  $p$  dalam  $P$ . Selanjutnya,

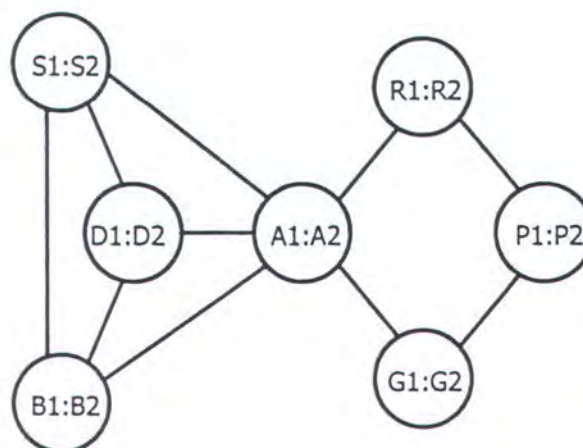


hubungkan dua *node*  $(v_1, v_2)$  dan  $(v'_1, v'_2)$  dari  $G$  yang menunjukkan keadaan yang *compatible* dengan  $R$  (contoh  $r(v_1, v'_1) = r(v_2, v'_2)$ ).

Sebagai contoh menggambar suatu "*association graph*", yang mempertimbangkan dua *layout catering deck* kapal penumpang yang pola topologinya dicocokkan (lihat gambar 3.3).



Gambar 3.3. Dua layout sederhana untuk pembuatan suatu *association graph*.



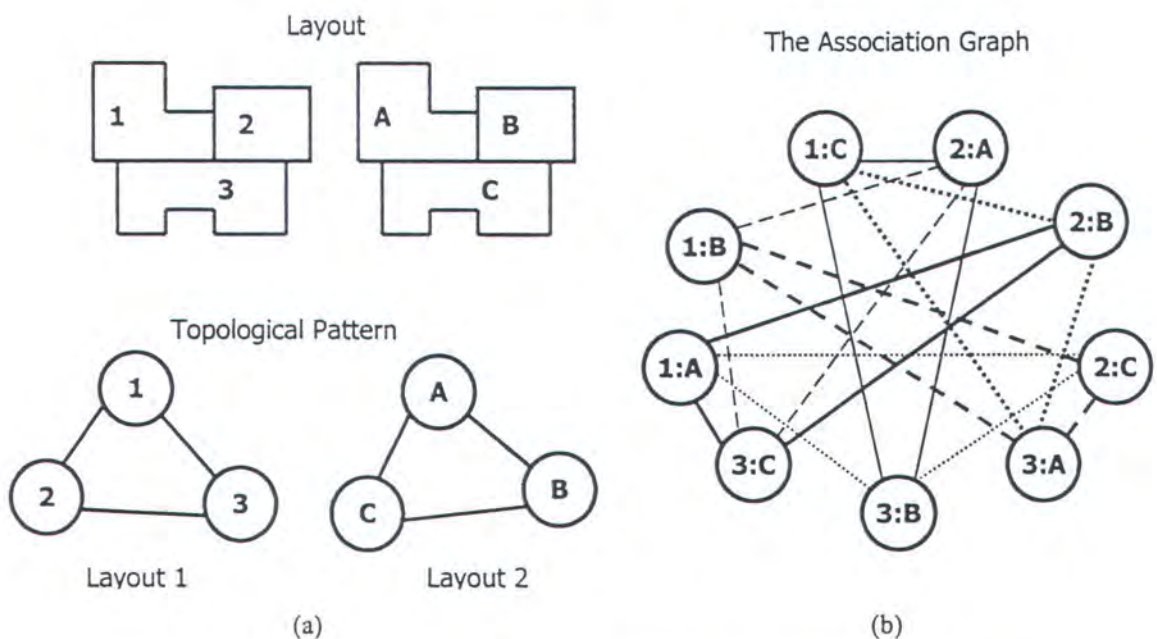
Gambar 3.4. Suatu *association graph* yang dihasilkan dari *matching* dua layout pada gambar 3.3.



### 3.4. Geometric Pattern Matching

Pencocokan bentuk geometrik dari layout tidak hanya bentuk geometriknya saja, tetapi ditinjau dulu pola topologi antar ruang dari masing-masing layout yang dicocokkan tersebut (Manfaat dkk., 1998). Sehingga pendekatan yang digunakan adalah perpaduan antara *association graph technique* untuk *topological pattern matching* dengan metode *planar shape matching*. Terdapat dua tahap dalam perpaduan *matching* tersebut, yaitu:

- pencocokan pola topologi dari layout, dan
- pencocokan bentuk dari beberapa *nodes* (ruang) yang berkorespondensi dengan metode *planar shape matching*. Hasil dari pencocokan bentuk dari setiap *nodes* yang berkorespondensi diakumulasi untuk mengukur kemiripan bentuk dari keseluruhan antar layout.



Gambar 3.5. (a) Dua layout sederhana yang identik untuk *geometric shape matching*  
(b) *Association graph* dari dua layout.

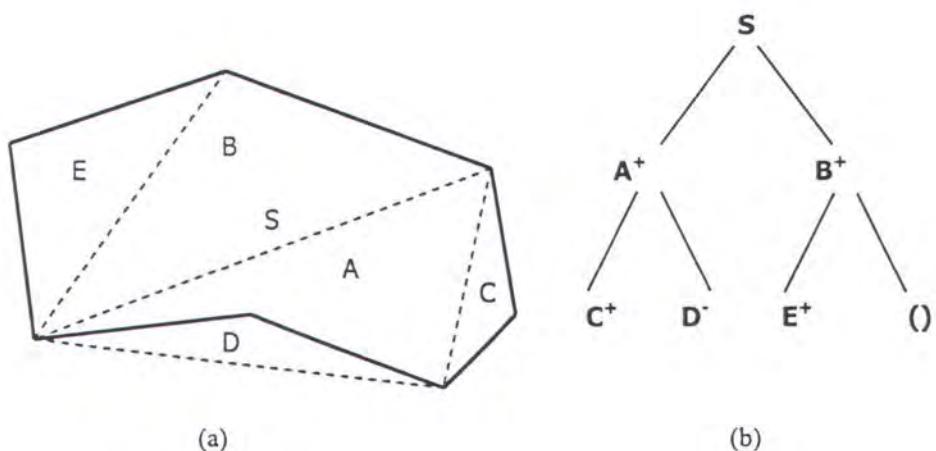


### 3.5. Representasi Bentuk

Untuk mempermudah dalam pemecahan permasalahan, dalam *shape matching technique*, bentuk ruangan diasumsikan:

- bentuk ruang adalah planar,
- garis batasnya tertutup, dan
- pada garis batas tersebut tidak ada yang saling memotong.

Pada *shape matching* bentuk ruangan direpresentasikan dalam suatu *binary tree structure* (Leu dan Huang, 1988). Dimana nodes menunjukkan segitiga dari pecahan bentuk yang dihasilkan. Suatu *binary tree* adalah suatu pohon (*tree*) yang setiap node-nya mempunyai dua *node* yang lain yaitu *node* anaknya (*children*). Lebih Jelasnya dapat dilihat pada gambar 3.6.



Gambar 3.6. Partisi bentuk: (a) Suatu bentuk terpartisi dalam segitiga; (b) Hasil *binary tree* dari partisi pada gambar (a) (Leu dan Huang, 1988)



Dari *binary tree* pada *node*-nya terdapat  $+$ , tanda  $-$  dan  $()$ , masing-masing mempunyai arti tanda  $+$  adalah bagian dalam (bagian dari bentuk), tanda  $-$  berarti bagian diluar bentuk, sedangkan tanda  $()$  berarti kosong.

Setiap *node* mempunyai tiga atribut, yaitu:

1. Luasan segitiga,
2. Rasio antara tinggi segitiga  $h$  dengan alas segitiga  $a$  ( $h/a$ ), dan
3. Rasio antara proyeksi dari sisi kiri segitiga terhadap alasnya  $b$  dengan alasnya ( $b/a$ ).

Atribut diatas membuat representasi bentuk *view-independent*. Hal ini karena mereka tidak tergantung dari lokasi dimana segitiga tersebut ditampilkan.

### 3.6. Penyusunan Data

Data Tugas Akhir ini data yang digunakan adalah data ruang akomodasi kapal penumpang. Masing-masing kapal diambil empat geladak yang dianggap memiliki karakteristik yang sama, sehingga data dikelompokkan menjadi empat berdasarkan geladak tersebut. Jadi data tersebut terbagi menjadi empat kelompok layout, yaitu Deck\_1, Deck\_2, Deck\_3 dan Deck\_4. Masing-masing kelompok data memiliki jumlah data sebanyak jumlah data kapal. Data dari tiap geladak merupakan data yang berdiri sendiri dan tidak ada hubungan satu sama lain dalam proses matching untuk geladak yang berbeda. Dan masing-masing layout mempunyai beberapa ruangan dimana masing-masing ruang mempunyai atribut sendiri-sendiri.



### 3.7. Prosedur Pengolahan Data

Data diolah dengan prosedur sebagai berikut:

1. Data didefinisikan dalam suatu kelompok data (*class*)

Tujuan dari proses pendefinisian ini adalah untuk mengelompokkan data dalam satu bagian yang dapat diakses oleh program pengakses. Dalam *class* data-data input dibagi kedalam tiga kategori, yaitu *space name*, *space function* dan *space identifier*.

Ketiga kategori ini harus dimasukkan sebagai input untuk setiap ruangan. *Space name* merupakan nama dari masing-masing ruangan. *Space function* merupakan kegunaan dari setiap ruang. Sedangkan *space identifier* merupakan keterangan dari input data yang ditentukan oleh desiner, apakah data tersebut masuk dalam kategori *user input* atau data yang dimasukkan dalam *case base*. Seluruh data harus memiliki kesamaan nama, fungsi dan identifikasi untuk ruangan dengan kategori yang sama dari kapal yang berbeda. Hal tersebut dapat dilihat pada contoh berikut:

```
(Defrooms :space-name 'Gangway1-1-01s
           :space-function 'Access
           :space-identifier 'Stored-Object)
```

Data yang digunakan dibagi menjadi empat *class* yang merupakan representasi dari empat geladak yaitu Deck\_1, Deck\_2, Deck\_3 dan Deck\_4. Faktor yang harus diperhatikan adalah setiap kelas tidak saling berhubungan dalam *graph*, sehingga proses *matching* terjadi hanya dalam masing-masing *class*.





## 2. Klasifikasi data dalam *case base*

Dalam proses matching ada dua macam input data yang dikenal dan digunakan, yaitu input eksternal dan input internal. Input eksternal biasa dikenal sebagai *user input*, dimana data input diberikan oleh perancang. Sedangkan input internal merupakan data yang sudah tersimpan dalam *case base*. *User input* dapat disimpan dalam internal input, dengan cara membagi data dalam *template* dan *stored-object*.

*Stored-object* adalah data yang tersimpan dalam *case base* dan berfungsi sebagai data pembanding dari *user input*. Data-data adalah kumpulan dari empat *class* yang sudah ditentukan sebelumnya, yaitu dari empat geladak yang ada.

*Template* adalah *stored-object* yang dimodelkan sebagai *user input* dan digunakan sebagai dasar dari pengenalan pola yang berarti input pola ke dalam *pattern matching process* (Hall dan Matias, 1993).

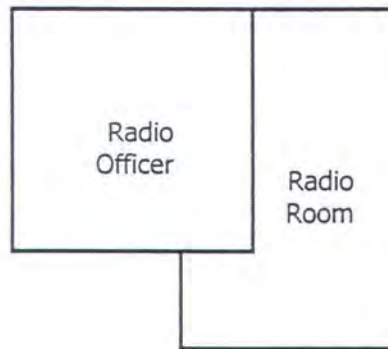
Cara pembagiannya adalah dengan menentukan pilihan pada *space identifier* yang dapat berupa *template* atau *stored-object*. Jika *space identifier* dinyatakan sebagai *template* maka data akan dikategorikan sebagai *user input* dan jika *space identifier* berupa *stored-object* maka data akan dikategorikan sebagai data pembanding yang dimasukkan dalam *case base*.

## 3. Kedekatan antar ruang (*space closeness*)

Berdasarkan teknik yang dipakai, yang dimaksud dengan kedekatan antar ruangan adalah bila dua ruangan berdampingan dan dipisahkan



oleh pembatas. Dalam graph kedekatan dilambangkan dengan garis yang menghubungkan antara dua ruang yang berdampingan. Jadi pada proses penyamaan (*matching*) dua layout maka harus memperhatikan hubungan ruangan yang berdekatan langsung. Sebagai contoh untuk dua ruang seperti gambar berikut:



Gambar 3.7. Gambar dua ruang yang didefinisikan kedekatannya.

Maka kedekatannya didefinisikan sebagai berikut:

```
(Defcloseness :name 'Deck-1-Ship-03  
:first-space 'Radio-Officer-1-03s  
:second-space 'Radio-Room-1-03s)
```

#### 4. Data *matching*

Data direpresentasikan dalam *graph* sebagai satu kesatuan yang terdiri dari *sub-graph* dan garis korespondensi. Setiap ruangan akan dinyatakan sebagai *sub-graph*. *Sub-graph* dinyatakan dengan *node* dan setiap ruangan yang berbatasan akan dihubungkan dengan sebuah garis korespondensi yang melambangkan kedekatan antar ruangan. Gabungan dari beberapa *sub-graph* dan garis korespondensi menjadi suatu *graph*. *Graph* inilah yang nantinya digunakan dalam proses *topological pattern*





*matching*. Dalam *sub-graph* tersebut terdapat data yang merepresentasikan suatu *binary tree* yang nantinya digunakan untuk proses *shape matching*.

### 3.8. Implementasi pendekatan *Case Based Reasoning in Design*

Sesuai dengan batasan masalah, pada tugas akhir ini pendekatan CBR *in Design* diimplementasikan sebatas penarikan (*recalling*) *case* pada *case base*, yaitu dengan menggunakan teknik *geometric and topological pattern matching*. Dimana teknik-teknik tersebut telah dijelaskan pada sub-bab-sub-bab sebelumnya.

Sebagai data dalam *case base* adalah layout ruang akomodasi ABK dari kapal-kapal penumpang dengan kapasitas 1000 orang. Data tersebut disimpan dalam *case base* dalam bentuk koordinat dari ruangan-ruangan penyusun layout beserta properties yang menunjukkan ciri dari ruangan tersebut, data kedekatan yang menunjukkan topologi dan data ruangan tersebut yang disusun dalam layout.

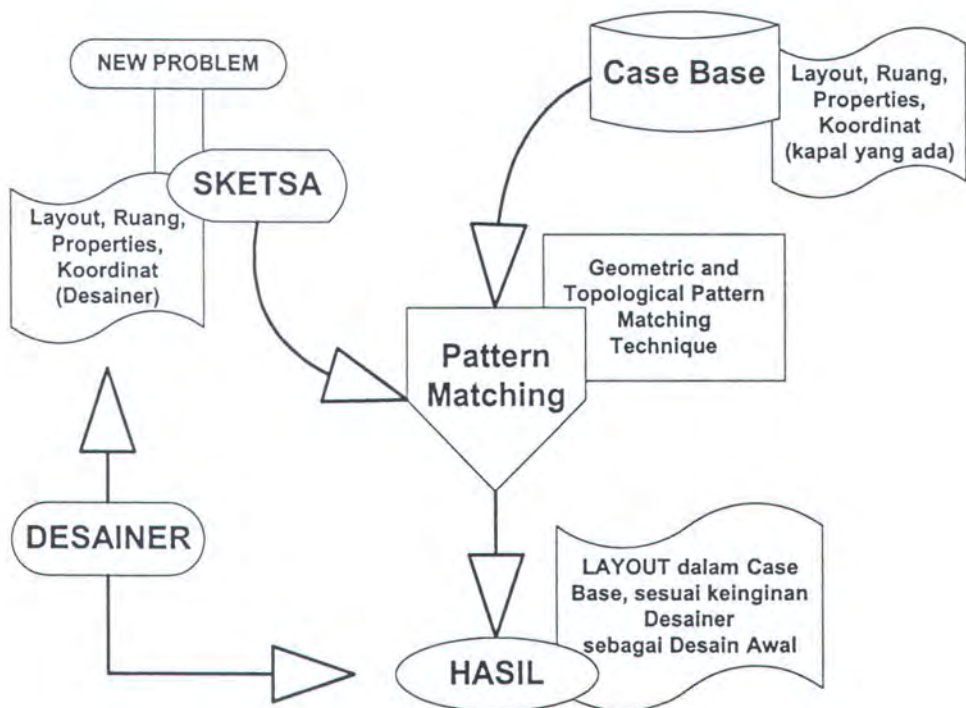
Proses penarikan dilakukan dengan mengacu pada kesamaan topologi dan bentuk dengan teknik *association graph matching* untuk memperoleh kesamaan topologi dan teknik *planar shape matching* untuk memperoleh kesamaan geometrik. Dengan input adalah suatu sketsa layout., dimana sebagai data input sama dengan data yang digunakan sebagai data cases.

Hasil yang akan diperoleh adalah data layout dengan urutan data yang mempunyai kesamaan tertinggi dengan sketsa layout sampai data yang





mempunyai tingkat kesamaan terendah. Hasil inilah yang nantinya dijadikan desain awal oleh desainer, dimana desainer bebas memilih layout mana yang diambil. Jadi implementasi teknik-teknik diatas mempunyai manfaat bagi desainer menunjukkan acuan atau memilihkan desain yang sesuai dengan keinginannya. Hal ini karena semakin layout tersebut mempunyai tingkat kemiripan yang tinggi, maka layout tersebut semakin mendekati keinginan desainer.

Gambar 3.8. Diagram *recalling process*



# BAB IV

## SISTEM SPACEMAN







## BAB IV

### SISTEM SPACEMAN

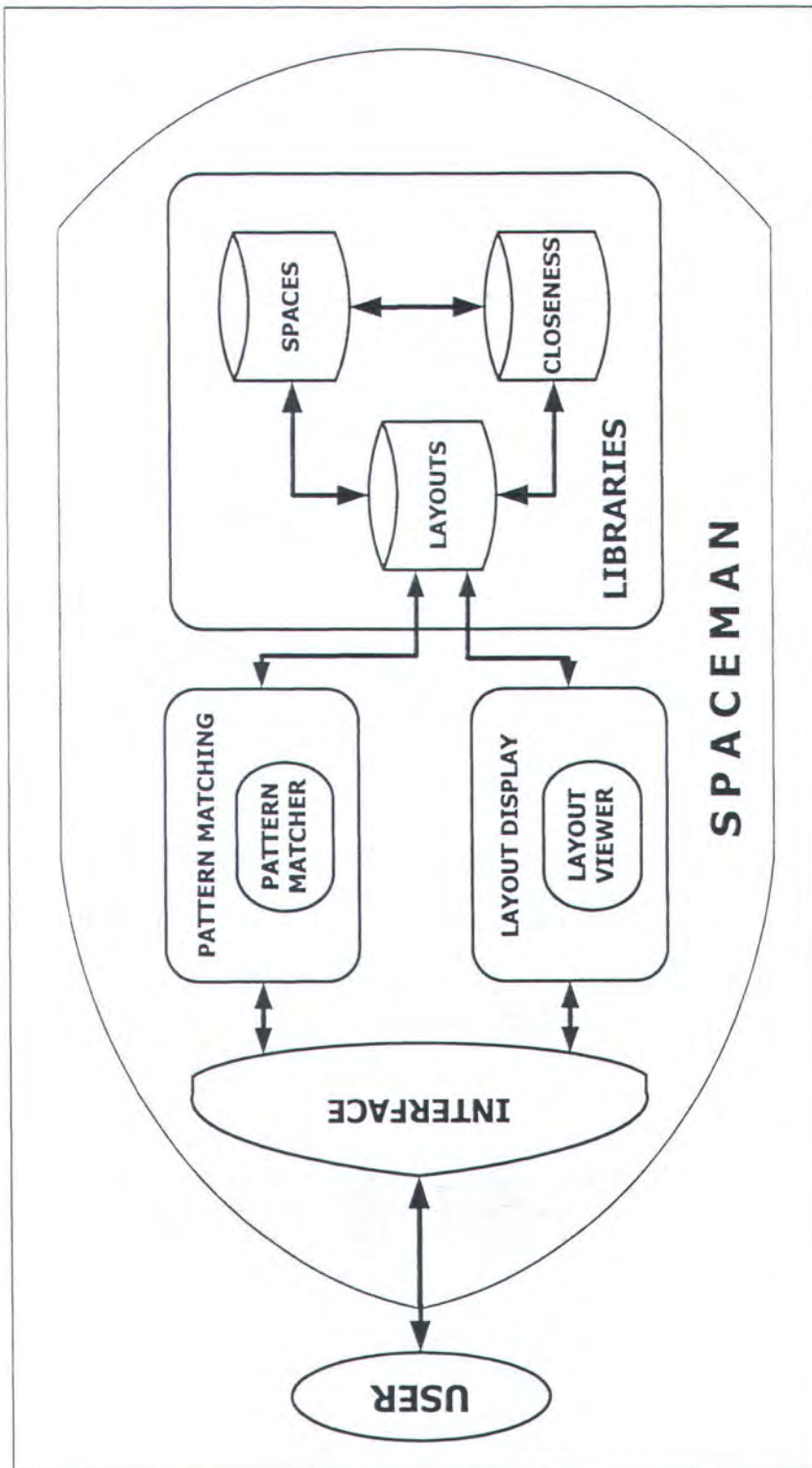
#### 4.1. Sistem SpaceMan

Teknik penarikan case dalam case based dengan mengacu pada kecocokan bentuk dan topologi ruang suatu layout disusun dalam suatu program komputer dan dinamakan sistem *SpaceMan*. Dapat dikatakan bahwa sistem ini adalah implementasi pendekatan *Case Based Reasoning*, yang mendukung pemanfaatan kembali desain lama (*design reuse*) dengan teknik penyamaan *Geometric and Topological Pattern Matching*.

SpaceMan merupakan merupakan salah satu *tools* untuk melaksanakan *pattern matching* dari layout. Dengan diberikannya input layout, sistem ini akan melakukan proses pencarian desain lama, kemudian mencari kesamaan (*similarities*) antara desain input dengan desain-desain lama. Selanjutnya dipilih desain yang paling sesuai (*most similar*) dengan input layout.

Secara umum sistem ini ditunjukkan oleh gambar 4.1. Yaitu terdiri dari kumpulan data (*libraries*), pola penyamaan (*pattern matcher*), *layout viewer* dan tampilan window (*interface*).





Gambar 4.1. Arsitektur dari Sistem SpaceMan



#### 4.2. Libraries

Libraries terdiri atas data layout, data ruang dan data kedekatan antar ruang. Data layout terdiri dari kumpulan ruang., data ruang menunjukkan atribut yang *unique* dari tiap-tiap ruangan yang ada pada layout karena semua memiliki nama yang berbeda, sedangkan data kedekatan adalah data yang menunjukkan topologi dari ruangan-ruangan dalam suatu layout.

#### 4.3. Pola Penyamaan (*Pattern Matcher*)

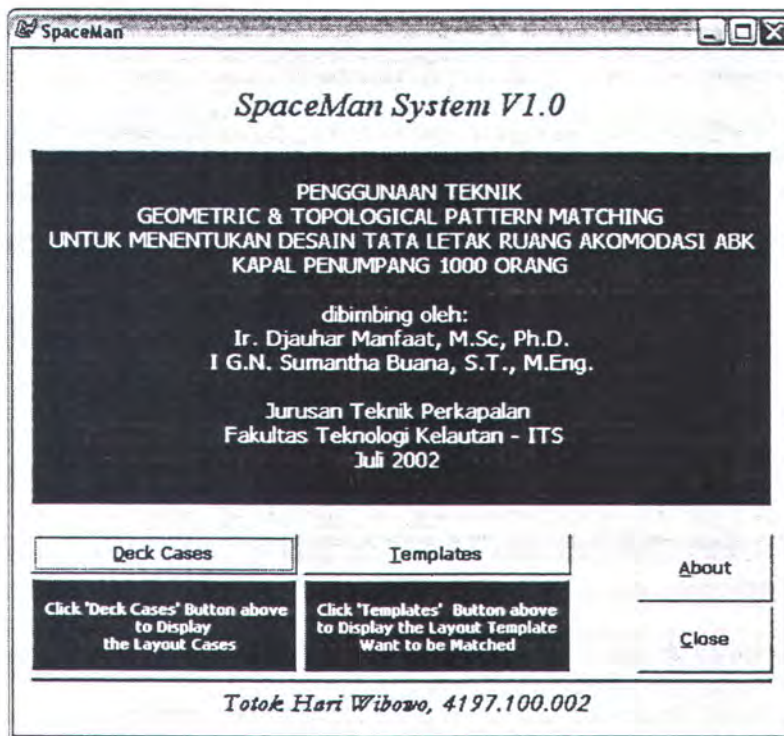
Pola penyamaan (*pattern matcher*) digunakan untuk penarikan desain lama jika diberikan suatu *input layout*. Dalam tugas akhir ini digunakan *graph matcher* digunakan sebagai *tools* untuk menyamakan dua bentuk layout dengan menggunakan metode *geometric shape matching* yang melibatkan teknik *topological pattern matching*. Dalam pattern matching jenis ini bentuk penyamaan (*matcher*) akan berjalan jika hubungan *pattern matching* dari layout telah lengkap.

#### 4.4. Layout Viewer

*Layout viewer* merupakan komponen dari sistem SpaceMan yang digunakan untuk menampilkan (*display*) bentuk dari layout desain lama maupun desain input dari *user*.



#### 4.5. Tampilan Window (Interface)



Gambar 4.2. Interface utama sistem SpaceMan

Sistem SpaceMan menyediakan fasilitas bagi user untuk melihat dan mengoperasikan sistem dalam suatu interface window yang sederhana, sehingga sangat mudah dioperasikan.

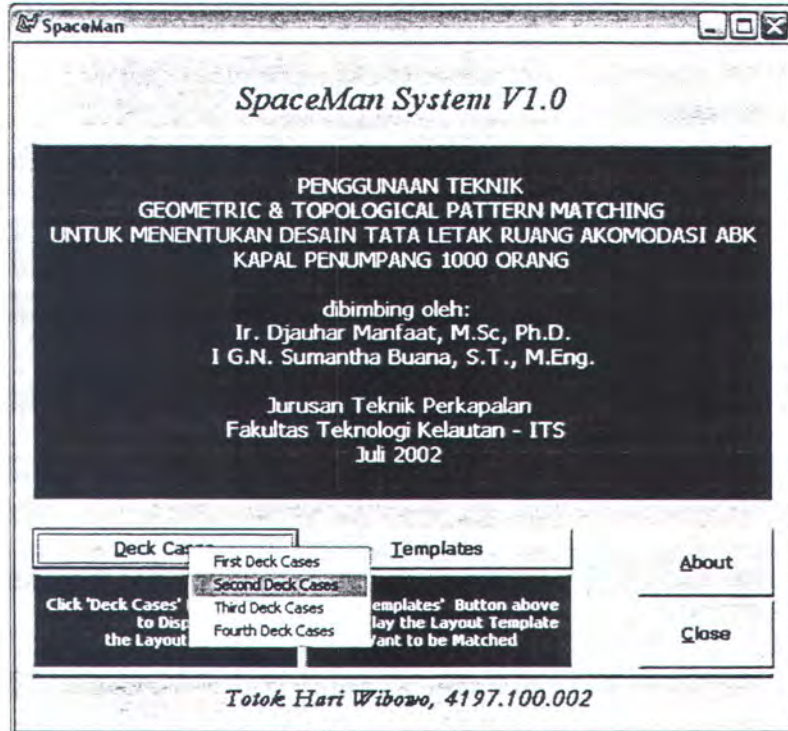
Dalam interface utama ini terdiri dari tiga tombol (*button*), yaitu:

1. Tombol **C**lose, berfungsi untuk menutup program.
2. Tombol **A**bout, untuk menampilkan penjelasan tentang sistem SpaceMan.
3. Tombol **D**eck Cases, untuk menampilkan data-data (layout cases) yang ada pada sistem ini, dan akan muncul empat kelompok data sesuai dengan lokasi (pada geladak mana layout itu berada) dari layout.



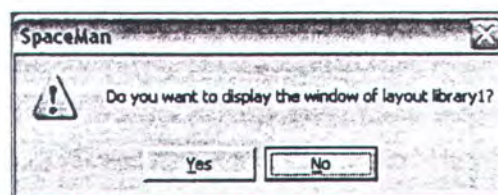


Jika tombol ini ditekan, akan menampilkan menu pilihan kelompok layout data seperti gambar berikut:



Gambar 4.3. Menu pilihan dari tombol Deck Cases.

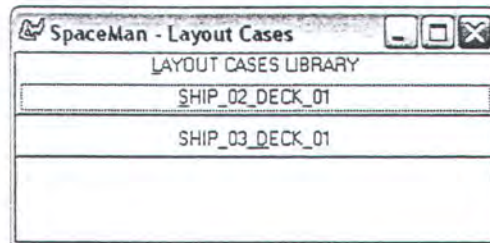
Terdapat empat pilihan yaitu **First Deck Cases**, **Second Deck Cases**, **Third Deck Cases** dan **Fourth Deck Cases**, yang masing masing terdiri dari beberapa layout dengan karakteristik yang sama. Misalnya dipilih (*Click*) **First Deck Cases**, akan tampil pilihan layout untuk ditampilkan dilayar. Yang sebelumnya akan tampil dialog sebagai berikut:



Gambar 4.4. Dialog untuk menampilkan pilihan layout

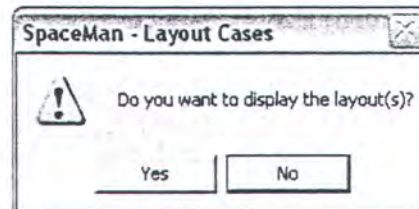


Jika dipilih tombol **Yes** maka akan tampil pilihan layout sebagai berikut:



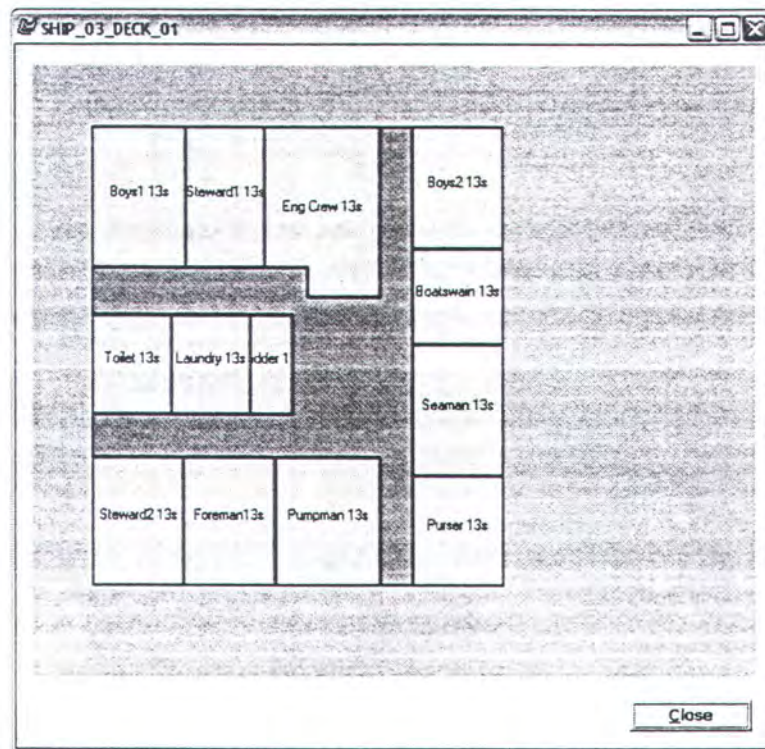
Gambar 4.5. Pilihan layout untuk semua data layout pada geladak pertama.

Ditunjukkan ada empat buah data layout. Dengan memilih salah satu akan ditampilkan gambar layoutnya, dan sebelumnya akan muncul dialog lagi.



Gambar 4.6. Dialog untuk mrnampilkan gambar layout

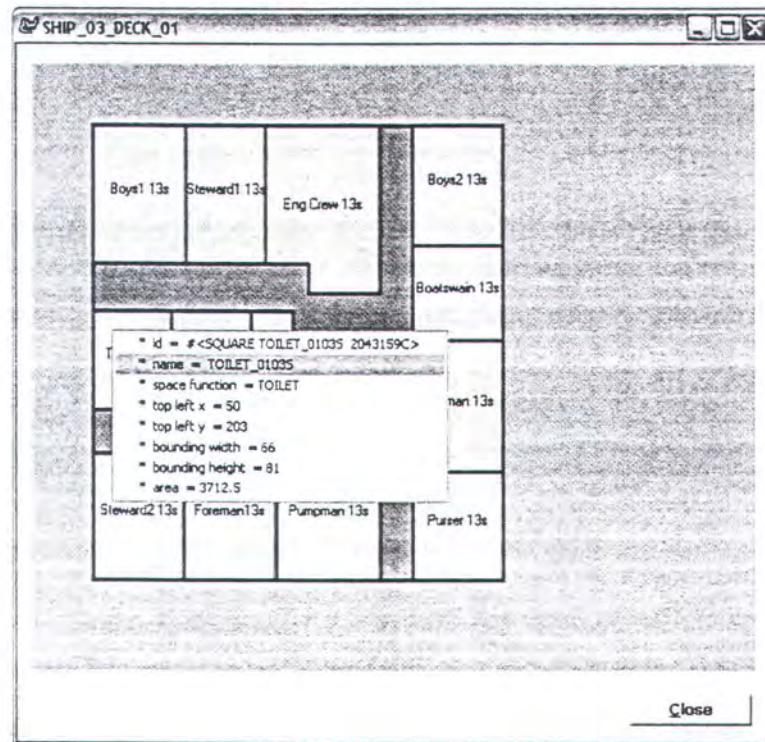
Selanjutnya akan muncul gambar dari layout yang dipilih.



Gambar 4.7. Gambar layout yang telah dipilih

Dalam tampilan window gambar layout di atas terdapat tombol Close digunakan untuk menutup window tersebut. Selain atribut dari masing-masing ruangan dalam layout tersebut dapat ditampilkan dengan menekan tombol kanan mouse pada posisi pointer di atas ruang yang ingin dilihat atributnya.

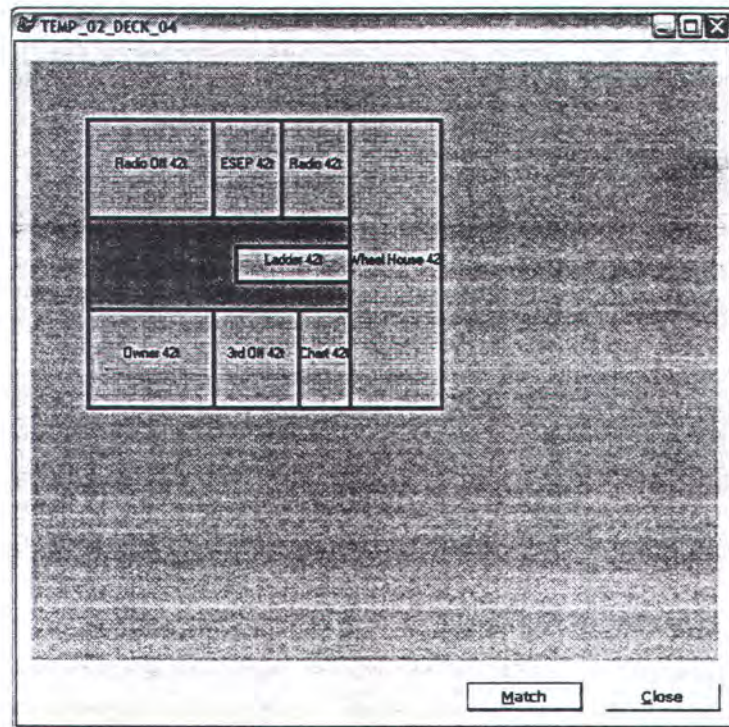




Gambar 4.8. Tampilan atribut untuk Toilet 13s.

4. Tombol **Template**, akan menampilkan *layout input* untuk proses *pattern matching* yang dinamakan data *template* (Hall dan Matias, 1993).

Tidak jauh berbeda dengan tombol **Deck Cases**. Hanya saja jika ditelusuri sampai tampilan window layout, pada tampilan window layoutnya terdapat tombol **Match**, yang berfungsi untuk mengaktifkan *pattern matcher* untuk melakukan tugasnya mencari layout yang mempunyai tingkat kemiripan tertinggi dengan layout yang ditampilkan pada data layout yang ditampilkan oleh tombol **Deck Cases**.



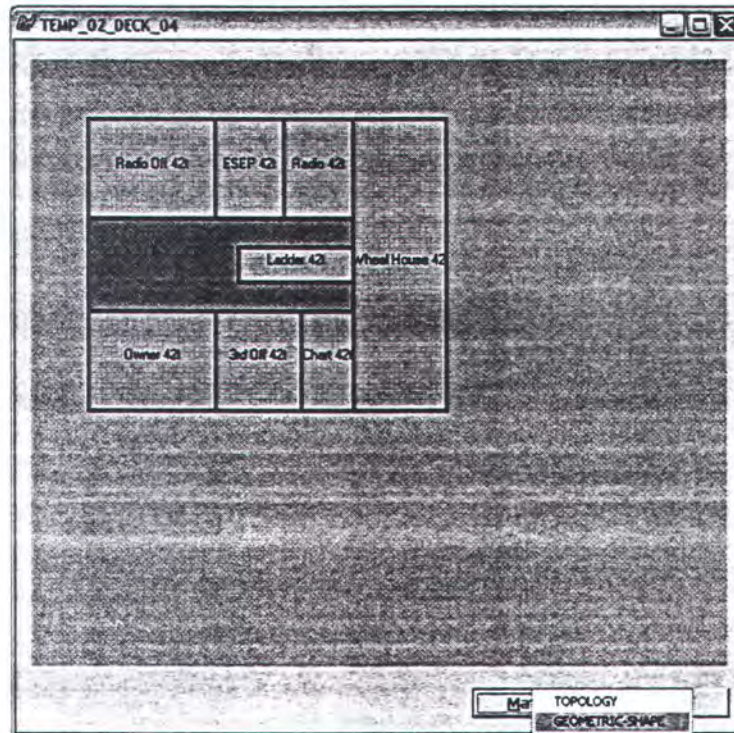
Gambar 4.9. Gambar layout template yang telah dipilih, siap untuk diproses.

Jika ditekan tombol **Match** akan muncul dua pilihan yaitu **TOPOLOGY** dan **GEOMETRIC-SHAPE** yang merupakan pilihan untuk proses *pattern matching*. Jika seperti dipilih **TOPOLOGY** maka proses *matching* akan menggunakan metode *topological pattern matching* yaitu menggunakan teknik *association graph matching* untuk penyamaan antar dua layout. Sedangkan **GEOMETRIC-SHAPE** dipilih maka proses *matching* akan menggunakan metode *geometric and topological pattern matching* yang menggunakan teknik kombinasi antara *association graph matching* untuk penyamaan antar dua layout dan *planar shape matching* untuk penyamaan dua ruang





yang berkorespondensi pada kedua layout tersebut. Pilihan tersebut ditunjukkan gambar berikut:



Gambar 4.10. Pilihan proses *matching* pada suatu template.

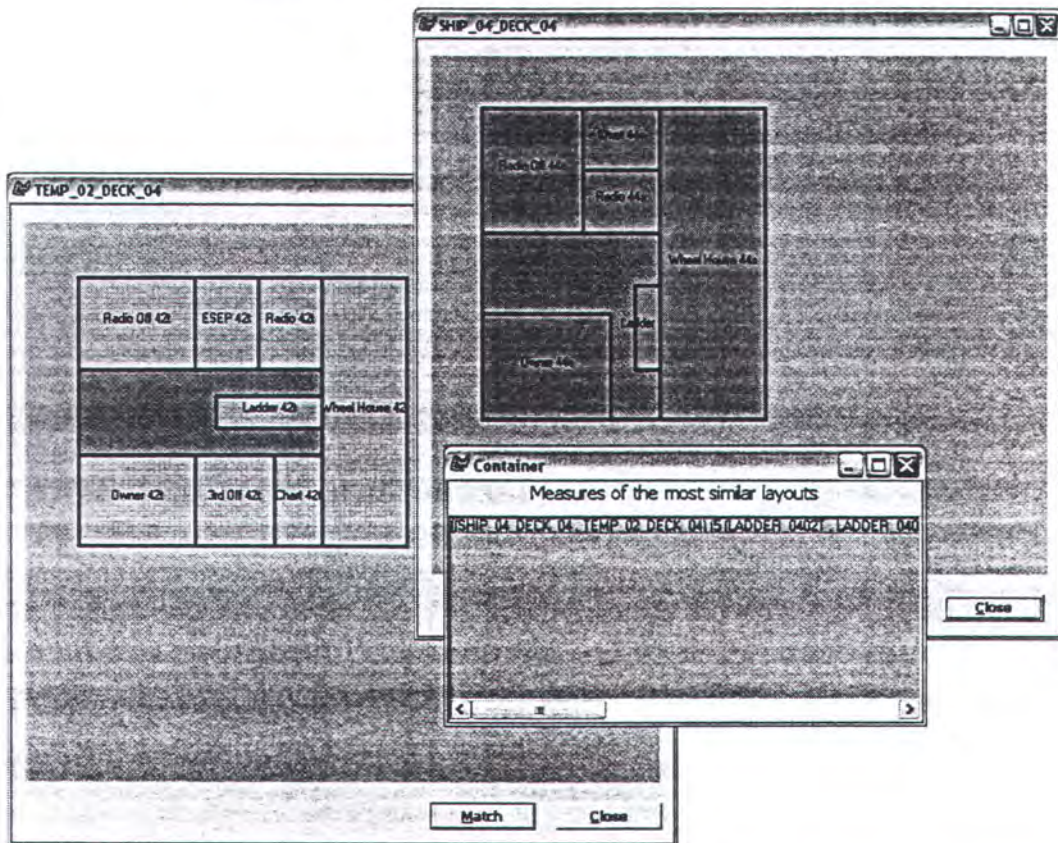
a. Pilihan **TOPOLOGY**

Seperti diuraikan diatas pilihan ini akan menghasilkan layout dengan metode *topological pattern matching*. User akan diberi pilihan apakah ingin mendapatkan hasil layout dengan tingkat kesamaan yang paling tinggi atau menarik layout dengan jumlah tertentu. Jika memilih dengan jumlah tertentu maka user akan memperoleh layout dengan urutan tingkat kesamaan yang paling tinggi sampai dengan tingkat kesamaan terendah sejumlah yang ia inginkan tadi. Untuk pilihan dengan layout yang mempunyai





kesamaan tertinggi hasilnya ditunjukkan pada gambar 4.11 berikut:



Gambar 4.11. Contoh hasil proses pilihan **TOPOLOGY** dengan hasil layout yang mempunyai tingkat kesamaan tertinggi secara topologi.

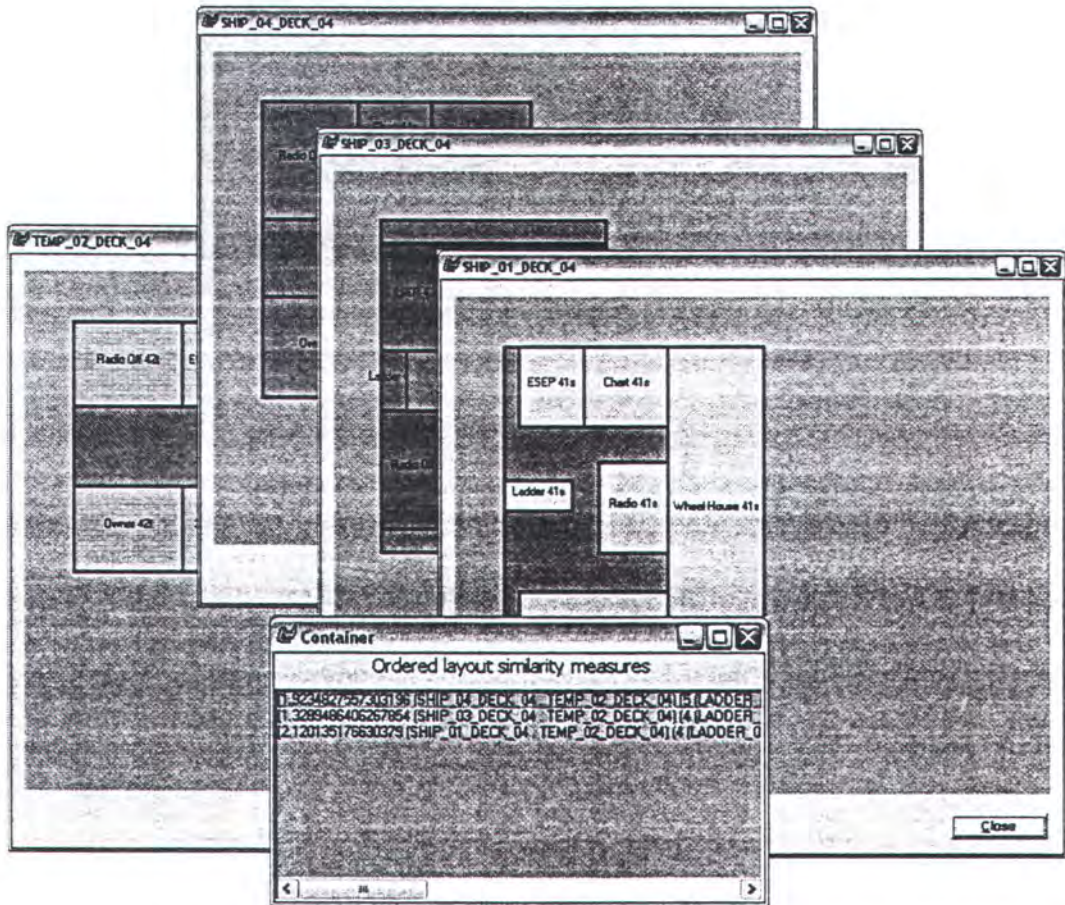
#### b. Pilihan **GEOMETRIC-SHAPE**

Pilihan ini adalah inti dari sistem SpaceMan. Seperti halnya dengan pilihan **TOPOLOGY**, pilihan **GEOMETRIC-SHAPE** ini akan memberikan hasil layout yang mempunyai tingkat kesamaan tertinggi atau beberapa layout dengan user input dan hasilnya adalah urutan dari layout yang mempunyai tingkat kesamaan tertinggi sampai terendah sejumlah yang diinput oleh user. Perbedaan yang akan muncul adalah pada *list* yang





dihasilkan *window* bernama *container*. Perbedaan itu adalah adanya *score* untuk hasil proses *geometric shape matching*. Berikut adalah contoh hasil *running* jika user input adalah untuk tiga layout yang mempunyai kesamaan tertinggi.



Gambar 4.12. Contoh hasil proses pilihan **GEOMETRIC-SHAPE** dengan hasil tiga layout yang mempunyai tingkat kesamaan tertinggi secara geometrik dan topologi.

Dalam *window container* terlihat hasil penarikan layout dari *case base*. Disana ditunjukkan urutan hasil layout dari layout yang mempunyai kesamaan tertinggi sampai terendah. *Score*, jumlah *clique* nama layout dan nama-nama ruang yang berkorespondensi dapat dilihat pada *window* tersebut.





## BAB V ANALISIS HASIL







## BAB V

### ANALISIS HASIL

Untuk keperluan analisis, digunakan data *template* yang secara *visual* mempunyai tingkat kesesuaian yang berbeda-beda sehingga dapat diketahui apakah hasil yang dicapai nantinya sesuai dengan hipotesa awal atau tidak. Setelah dilakukan *running* program untuk *matching* semua data *template*, hasil yang diperoleh ditabulasi sehingga mempermudah analisis. Tabel tersebut merupakan tabel kesesuaian yang berisi:

- urutan data layout kapal berdasarkan angka kesamaan hasil proses *matching* yang disusun berurutan dari *clique* tertinggi hingga terendah,
- angka kesamaan (*clique*) antara *layout template* dengan masing-masing data layout kapal,
- nama ruangan (*space*) hasil proses *matching* yang memiliki korespondensi (topologi), dan
- angka kesamaan (*score*) total semua *clique* antara *layout template* dengan masing-masing data layout kapal.

Dari proses *running* program, diperoleh hasil seperti pada tabel 5.1., tabel 5.2., tabel 5.3. dan tabel 5.4. masing-masing untuk hasil *matching* antara layout pada Template\_Deck\_4 dengan Ship\_Deck\_4, Template\_Deck\_3 dengan Ship\_Deck\_3, Template\_Deck\_2 dengan Ship\_Deck\_2 dan Template\_Deck\_1 dengan Ship\_Deck\_1.



Tabel 5.1. Hasil *matching* antara data layout Template\_Deck\_4 dengan Ship\_Deck\_4.

Nama Template Layout	Nama Ship Layout	Jumlah Clique	Ruangan Hasil Matching	Total Score
Temp_01_Deck_4	Ship_04_Deck_4	4	Radio Officer Room	1.44407
			Ladder	
			Chart Room	
			Gangway1	
	Ship_03_Deck_4	4	Radio Room	1.90767
			Ladder	
			Chart Room	
			Gangway1	
	Ship_01_Deck_4	4	Radio Room	1.92188
			Ladder	
			Chart Room	
			Gangway1	
	Ship_02_Deck_4	3	Radio Room	3.81662
			Chart Room	
			Gangway1	
Temp_02_Deck_4	Ship_04_Deck_4	5	Ladder	1.92348
			Wheel House	
			Chart Room	
			Owner Room	
			Gangway1	
	Ship_03_Deck_4	4	Ladder	1.32895
			Wheel House	
			Chart Room	
			Gangway1	
	Ship_01_Deck_4	4	Ladder	2.12014
			Chart Room	
			Radio Room	
			Gangway1	
	Ship_02_Deck_4	4	Chart Room	2.91913
			3rd Officer Room	
			Radio Officer Room	
			Gangway1	





Tabel 5.2. Hasil *matching* antara data layout Template\_Deck\_3 dengan Ship\_Deck\_3.

Nama Template Layout	Nama Ship Layout	Jumlah Clique	Ruangan Hasil Matching	Total Score
Temp_02_Deck_3	Ship_02_Deck_3	7	1st Officer Room	3.47964
			1st Engineer Room	
			Ladder	
			Steward Room	
			Captain Office	
			Captain Room	
			Gangway1	
	Ship_03_Deck_3	6	1st Engineer Room	1.87680
			Ladder	
			Captain Office	
			Captain Room	
			Chief Officer Room	
			Gangway1	
	Ship_04_Deck_3	6	1st Officer Room	2.55668
			1st Engineer Room	
			Ladder	
			Captain Office	
			Captain Room	
			Gangway1	
	Ship_01_Deck_3	5	1st Officer Room	1.95639
			1st Engineer Room	
			Ladder	
			Captain Room	
			Gangway1	



Tabel 5.3. Hasil *matching* antara data layout Template\_Deck\_2 dengan Ship\_Deck\_2.

Nama Template Layout	Nama Ship Layout	Jumlah Clique	Ruangan Hasil Matching	Total Score
Temp_01_Deck_2	Ship_01_Deck_2	10	Toilet	0.00763
			Musholla	
			Chief Cook Room	
			Oiler Room	
			Hospital	
			Ladder	
			Pantry	
			Messroom	
			Assistant Cook Room	
			Gangway1	
	Ship_04_Deck_2	6	Toilet	1.85480
			Chief Cook Room	
			Ladder	
			Pantry	
			Messroom	
			Gangway1	
			Chief Cook Room	
			Hospital	
			Ladder	
			Assistant Cook Room	
	Ship_03_Deck_2	5	Gangway1	2.75871
			Chief Cook Room	
			Hospital	
			Ladder	
			Assistant Cook Room	
	Ship_02_Deck_2	3	Gangway1	1.95151
			Ladder	
			Messroom	
			Gangway1	





Tabel 5.4. Hasil *matching* antara data layout Template\_Deck\_1 dengan Ship\_Deck\_1.

Nama Template Layout	Nama Ship Layout	Jumlah Clique	Ruangan Hasil Matching	Total Score
Temp_02_Deck_1	Ship_02_Deck_1	7	Ladder	1.61959
			Galley	
			Messroom	
			Boat Swain Room	
			Seaman Room	
			Boys Room	
			Gangway1	
	Ship_03_Deck_1	6	Ladder	2.75754
			Toilet	
			Boat Swain Room	
			Seaman Room	
			Boys Room	
			Gangway1	

Bila kita tinjau tabel 5.3, disana ditunjukkan hasil *matching* antara layout Temp\_01\_Deck\_2 dengan cases layout geladak kedua. Template ini sengaja dibuat sama secara visual dengan Ship\_01\_Deck\_2, dan hanya ada sedikit perbedaan ukuran ruang, yaitu pada ladder dan gangway (perbedaan panjang 1 point). Secara topologi jelas menunjukkan hubungan yang sama. Secara logika, jika dilakukan *matching* maka *case* yang akan menempati urutan pertama kemiripannya adalah Ship\_01\_Dec\_2. Dan setelah dilakukan *running* program menunjukkan hasil sesuai dengan logika tadi dengan waktu yang amat singkat dan *score* mendekati nol serta jumlah *clique* sebanyak ruangan. Hanya dalam beberapa detik semua *case* telah ditampilkan sesuai dengan urutan kemiripan (*similarity*).



Pada tabel 5.1 untuk *matching* antara Temp\_01\_Deck\_4 dengan *cases* pada geladak keempat dan antara Temp\_02\_Deck\_4 dengan *cases* pada geladak keempat dimana masing-masing memiliki tiga *cases* yang memiliki tingkat kesamaan topologi yang sama (masing-masing bernilai empat *clique*). Menunjukkan bahwa jika hasil *matching* menunjukkan jumlah *clique* yang sama (secara topologi mempunyai tingkat kesamaan yang sama) maka tingkat kesamaan ditunjukkan oleh *score* dari kesamaan bentuk. Dimana *score* yang paling mendekati nol adalah yang memiliki tingkat kesamaan tertinggi.

Ada kasus yang menarik pada tabel 5.1 yang hanya menunjukkan perbedaan pada bentuk. Yaitu pada hasil *matching* antara Temp\_01\_Deck\_4 dengan *cases* pada geladak keempat dihasilkan dua layout yang mempunyai tingkat kesamaan topologi yang sama dengan jumlah *clique* empat, dimana masing-masing *clique* dari kedua layout hasil tersebut pada ruangan-ruangan yang mempunyai korespondensi yang sama (punya fungsi yang sama). Ruangan-ruangan tersebut adalah Radio Room, Chart Room, Ladder dan Gangway. Hal ini menunjukkan performa dari sistem SpaceMan dalam menangani penyamaan bentuk (*shape matching*).

Dari uraian diatas menunjukkan teknik *geometric and topological pattern matching* sangat baik dalam memberikan hasil dari proses *recalling* suatu *case* dari *case base*. Hal ini ditunjukkan dengan hasil urutan yang kemungkinan menduanya kecil, yang sangat penting sekali dalam pengambilan keputusan terhadap desain layout hasil *pattern matching*. Dengan menggunakan komputer akan memberikan kecepatan yang tinggi dan pengurutan yang baik dari suatu





proses *recalling*. Sehingga dapat dikatakan bahwa sistem SpaceMan teruji dengan baik dalam melakukan proses *matching* dan mempunyai manfaat yang besar dalam proses desain layout.

Selanjutnya dalam hal ini desainer/user bebas memilih apakah yang dipakai adalah layout pada kapal yang mempunyai kesamaan tertinggi kemudian dilakukan modifikasi (*adaptation*) dari desain tersebut atau desainer mengambil yang mempunyai urutan ranking yang kedua atau desain yang lain. Atau desainer akan mengambil desain yang mempunyai nilai kesamaan yang tertinggi lalu akan mengambil bentuk-bentuk kesamaan-kesamaan dari desain-desain yang lain dan sebagainya. Dalam pemilihan desain tersebut desainer mempunyai pertimbangan-pertimbangan lain serta batasan-batasan tertentu yang dapat membantu desainer dalam mengambil keputusan.



**BAB VI**  
**KESIMPULAN DAN SARAN**





## BAB VI

### KESIMPULAN DAN SARAN

#### 6.1. Kesimpulan

Dari hasil perencanaan tataletak ruang akomodasi dengan sistem SpaceMan dapat ditarik kesimpulan sebagai berikut:

1. Pemanfaatan desain lama (*design reuse*) untuk membuat desain baru dapat membawa keuntungan bagi *desainer* dan *owner* (pemesan) karena desainer tidak perlu merencanakan desain dari awal (*from scratch*). Sehingga desain baru dapat dihasilkan dalam waktu yang lebih cepat dan lebih mendekati permintaan owner.
2. Hasil *matching* yang didapat dari system SpaceMan mempunyai kecepatan yang tinggi karena menggunakan komputer sebagai *tools*. Dengan demikian *desainer* dapat mendapatkan hasil desain yang sesuai dengan keinginan *owner* dengan cepat.

#### 6.2. Saran

Untuk memaksimalkan sistem SpaceMan perlu diperhatikan jumlah data yang digunakan. Semakin banyak data yang digunakan akan menghasilkan pilihan desain yang lebih variatif dan lebih mendekati keinginan owner.

Sistem ini juga bisa dikembangkan untuk membantu merancang layout selain ruang akomodasi ABK kapal penumpang, dengan cara

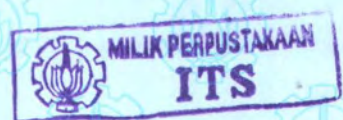


penambahan *option* jenis-jenis layout berdasarkan tipe kapal dan data layout kapal yang dimaksud dalam *case base*.





# DAFTAR PUSTAKA





## DAFTAR PUSTAKA

- Aamodt, A. and Plaza, E. (1994). *Case Base Reasoning: Foundation Issues, Methodological and System Approach*. AICom – Artificial Intelligent Communication Press vol. 7:4, 39-59.
- Akin, O. (1978). How do architects design? In J.L. Latombe (Eds.), *Artificial Intelligent and Pattern Recognition in Computer Aided Design*, pp. 65-103. Nort-Holland.
- Apple, James M. (1990). **Tata Letak Pabrik and Pemindahan Bahan**, Edisi Ketiga, Penerbit ITB Bandung.
- Ballard, D. H. and C. M. Brown. (1982). *Computer Vision*. Englewood Cliffs, New Jersey: Prentice-Hall, Inc.
- Dave, B., G. Schmitt, B. Faltings, and I. Smith. (1994). Case Based Design in Architecture. In J. S. Gero and F. Sudweeks (Eds.), *Artificial Intelligence in Design '94*, The Netherlands, pp. 145-162. Kluwer Academic Publishers.
- Omeshek, E. and J. Kolodner. (1992). A Case-Based Design Aids for Architecture. In J. S. Gero and F. Sudweeks (Eds.), *Artificial Intelligence in Design '94*, Dordrecht, pp. 497-516. Kluwer Academic Publishers.
- Faltings, B. (1997). **Case Reuse by Model-Based Interpretation**. In M. Maher and P. Pu (Eds.), *Issues and Applications of Case-Based Reasoning in Design*. Hillsdale, N.J.: Lawrence Erlbaum Associates.





- Flemming, U., Z. Aygen, R. Coyne, and J. Snyder. (1997). **Case-Based Design in a Software Environment that Support the Early Phases**. In M. L. Maher and P. Pu (Eds.), *Issues and Applications of Case-Based Reasoning in Design*. Hillsdale, N.J.: Lawrence Erlbaum Associates.
- Giretti, A., Spalazzi, L. and Lemma, M. (1994). A. S. A. *An Interactive Assistant to Architectural Design*. **Artificial Intelligence in Design**, 93-108.
- Hua, K., B. Faltings, and I. Smith. (1996). CADRE: *Case-Based Geometric Design*. **Artificial Intelligence in Engineering** 10(2), 171-183.
- Hall, G. and Matias, A. (1993). **Rotation, Scale and Translation Invariant Template Matching on a Transputer Network**. *Microprocessors and Microsystems* 17(6), 333-340.
- Kolodner, J. L. (1993). *Case-Based Reasoning*. New York: Morgan Kaufmann.
- Maher, M. L., M. B. Balachandran, and D. M. Zhang. (1995). *Case-Based Reasoning in Design*. Mahwah, New Jersey: Lawrence Erlbaum Associates, Publisher.
- Manfaat, D. (1998). **Computer-Based Approach to Effective Utilisation of Spatial Layout Design Experience**, Thesis for the Degree of Doctor, University of Strathclyde, Glasgow, Scotland, UK.
- Manfaat, D., Duffy, A. H. B. and Lee, B. S. (1996). *Review of Pattern Matching Approaches*. *The Knowledge Engineering Review* 11(2), 161-189.
- Manfaat, D., Duffy, A. H. B. and Lee, B. S. (1998). *Topological Pattern and Geometric Shape Matching for Layout Case Retrieval*. *Artificial Intelligence in Design*, 327-344.



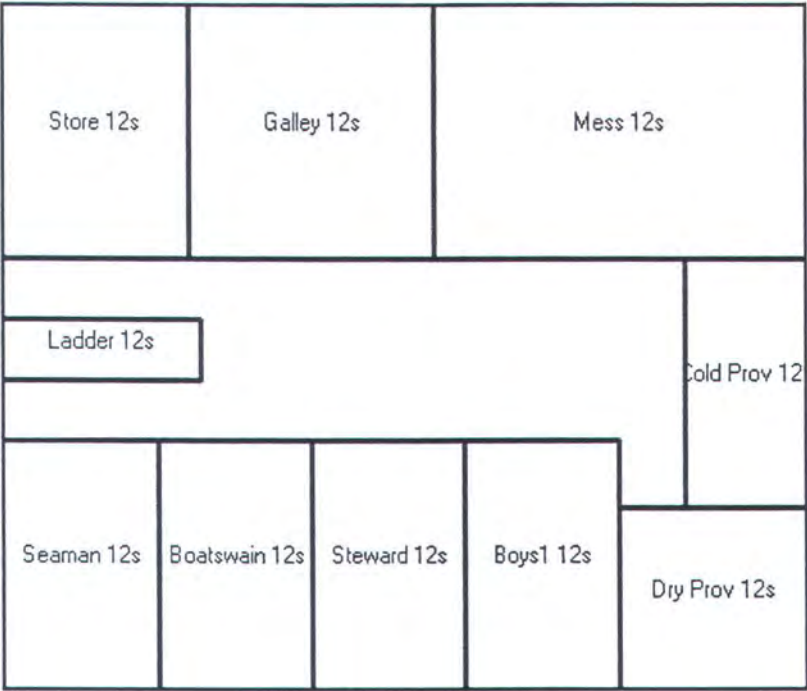
- Pangastuti, R. (2001). **Penerapan Teknik *Spatial Relation Ship Pattern Matching* dalam Pemanfaatan Desain Tataletak Ruang Akomodasi Anak Buah Kapal Penumpang 1000 Orang.** Jurusan Teknik Perkapalan, Fakultas Teknologi Kelautan – ITS.
- Purnatha, I G. (2000). **Analisis Pengambilan Keputusan Tataletak Ruang Akomodasi KM Caraka Jaya Niaga III,** Jurusan Teknik Perkapalan, Fakultas Teknologi Kelautan – ITS.
- Pusat Pembinaan dan Pengembangan Bahasa. (1990). **Kamus Besar Bahasa Indonesia.** Departemen Pendidikan dan Kebudayaan, Balai Pustaka, Jakarta.
- Sussman, Gerald J., Abelson, H. and Perlis, Alan J. (1996). ***Stricture and Interpretation of Computer Programs*,** Second Edition, The MIT Press, Cambrige, Massachusetts.



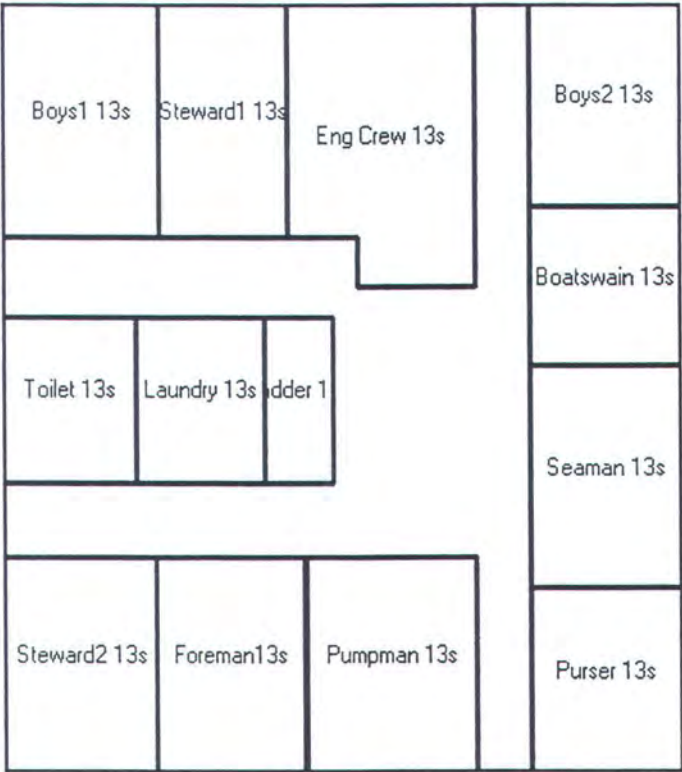


LAMPIRAN

LAMPIRAN



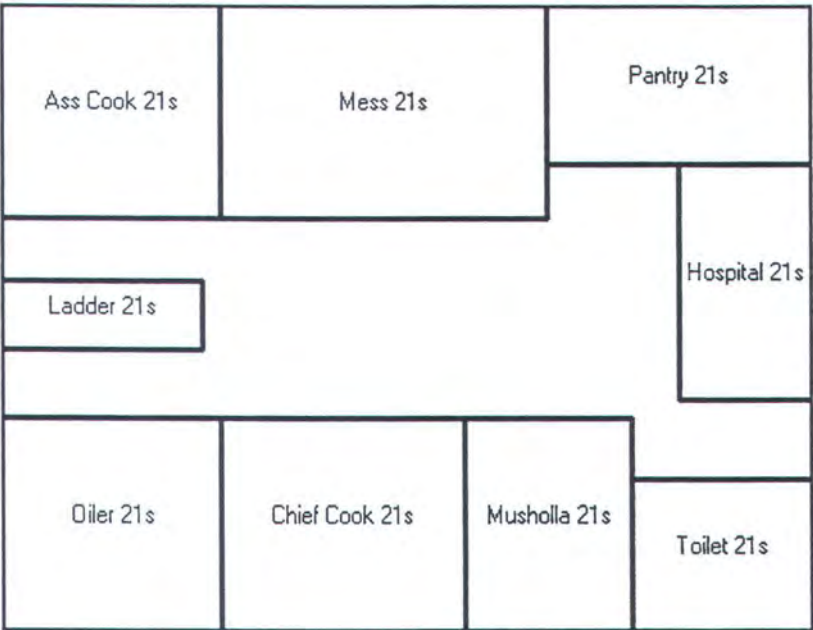
Layout Ship-02\_ Deck-1



Layout Ship-03\_ Deck-1



LAMPIRAN

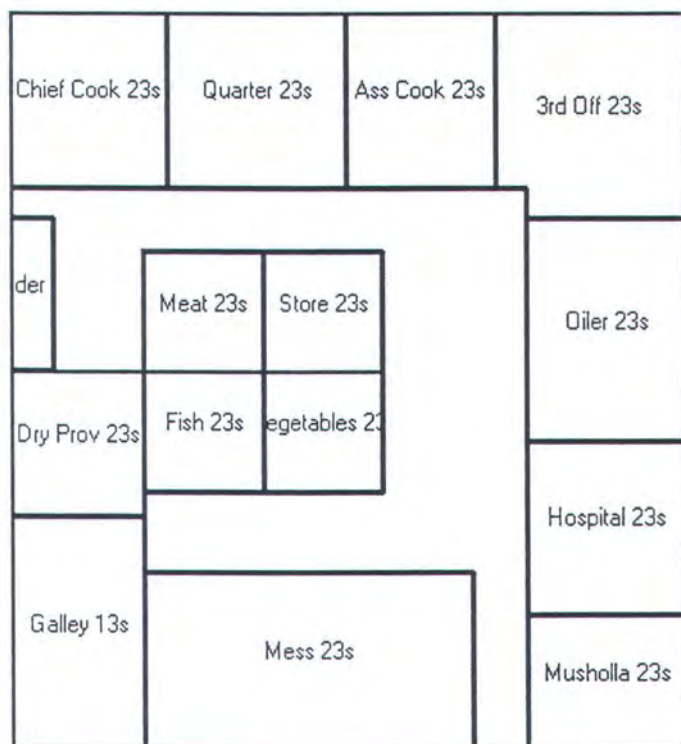


Layout Ship-01\_ Deck-2

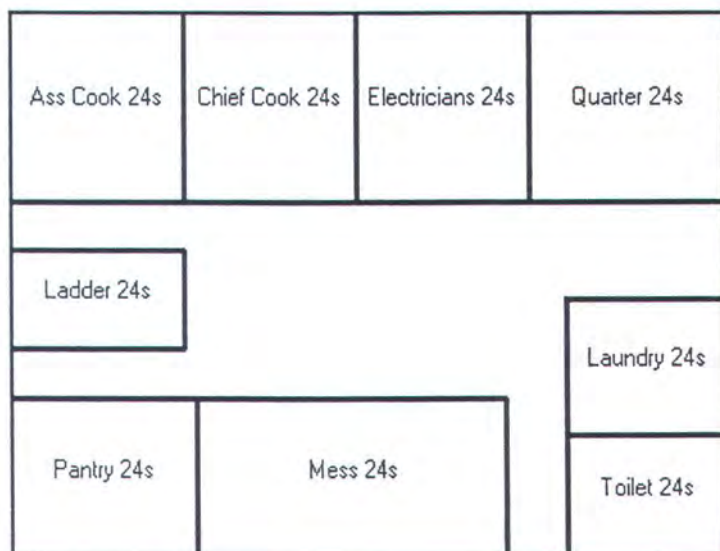


Layout Ship-02\_ Deck-2

## LAMPIRAN



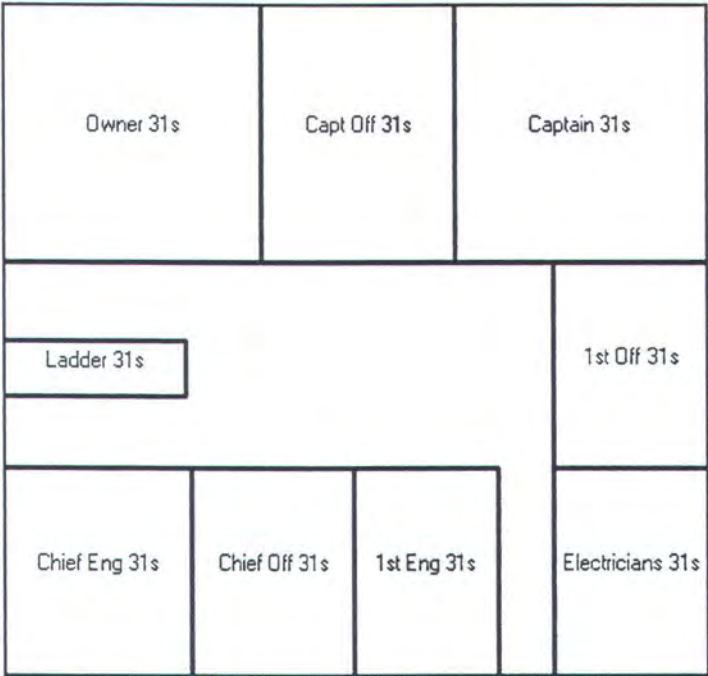
**Layout Ship-03\_ Deck-2**



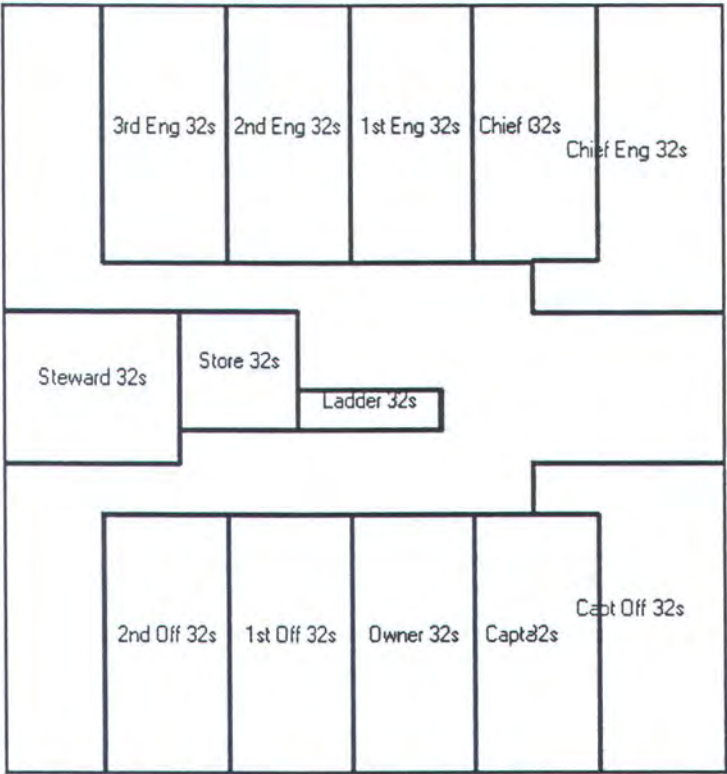
**Layout Ship-04\_ Deck-2**



LAMPIRAN



Layout Ship-01\_ Deck-3

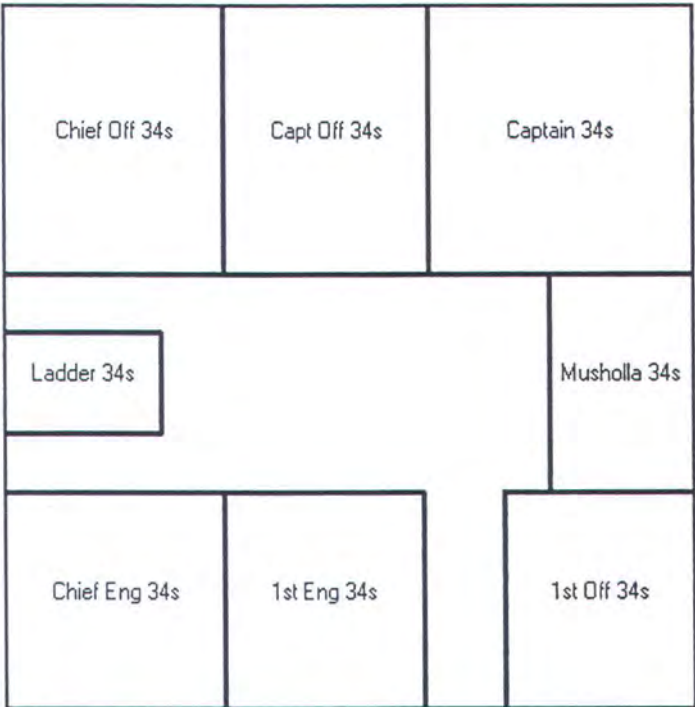


Layout Ship-02\_ Deck-3

LAMPIRAN



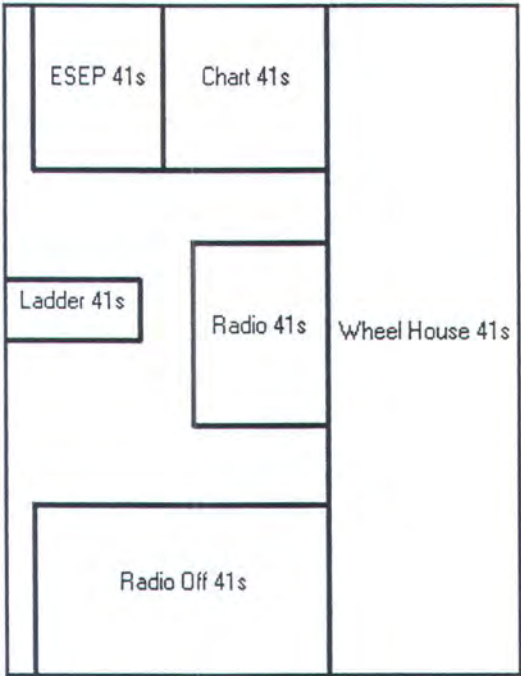
Layout Ship-03\_ Deck-3



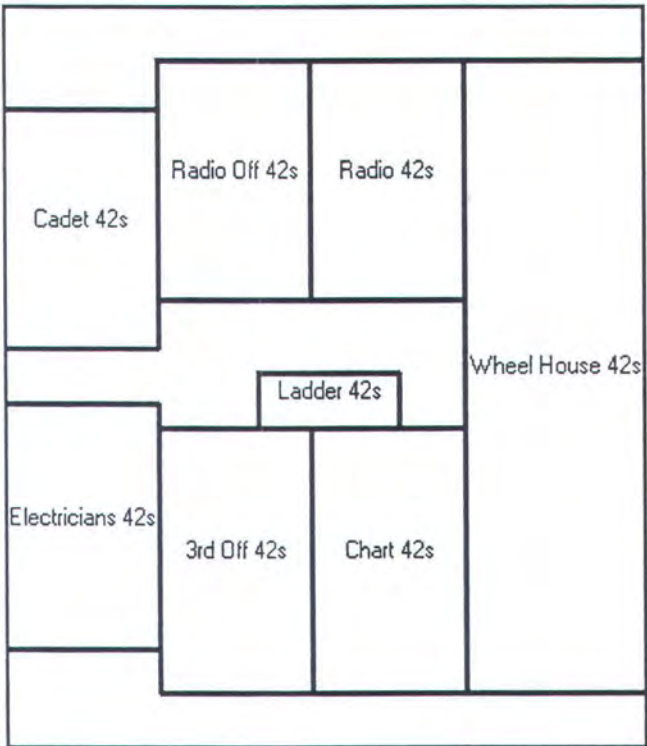
Layout Ship-04\_ Deck-3



LAMPIRAN



Layout Ship-01\_ Deck-4



Layout Ship-02\_ Deck-4

LAMPIRAN

